



University  
of Glasgow

<https://theses.gla.ac.uk/>

Theses Digitisation:

<https://www.gla.ac.uk/myglasgow/research/enlighten/theses/digitisation/>

This is a digitised version of the original print thesis.

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses

<https://theses.gla.ac.uk/>  
[research-enlighten@glasgow.ac.uk](mailto:research-enlighten@glasgow.ac.uk)

# **Modelling And Control Of Suspensions As Used In Interferometric Gravitational Wave Detectors**

David Andrew Palmer, MSci.,

Department of Mechanical Engineering

and

Department of Physics and Astronomy,

University of Glasgow.

Presented as a thesis for the degree of PH.D.,

University of Glasgow.

© David Andrew Palmer, 2001.

September 27, 2001



ProQuest Number: 10647745

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10647745

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

GLASGOW  
UNIVERSITY  
LIBRARY:

12634  
COPY 2

## **Abstract**

In this thesis the bond graph modelling of multi-stage pendula is presented. The models are abstractions of the real physical pendula as used in the GEO 600 gravitational wave detector. In addition, Model Based Observers (MBO) for the control of these real physical pendula, are presented.

The bond graph domain is utilised because it is an energy based methodology which facilitates the creation of unambiguous hierarchical models. This hierarchical property is fully exploited to produce a library of components which can be used to create complex multi-stage pendula in a modular fashion. Included in these models are hierarchical wire models which can model forces due to linear extension, bending dynamics and transverse modes of vibration. Components have been validated against a real physical system and as such alternatively configured system models can be created with confidence.

Bond graph pendulum models are incorporated in the design of Model Based Observer controllers. Model Based Observers are designed in the “physical domain” of bond graphs and as such provide an intuitive approach to controller design, and a unified approach to both system modelling and controller synthesis.

As a case study, a Model Based Observer has been designed and successfully implemented on a real physical pendulum system. This controller damps the resonant motion of a real double pendulum. A more sophisticated Model Based Observer, incorporating a split feedback topology, which is designed to control the position of a multi-stage pendula has also been investigated.

# Acknowledgements

Firstly I would like to thank both my supervisor Professor Peter Gawthrop, for affording me the opportunity to undertake the work detailed within this thesis, and the Engineering and Physical Sciences Research Council (EPSRC) for providing the necessary funds.

I would also like to thank Professor Norna Robertson, Doctor Donald Ballance and Doctor Ken Strain who have all, at various times, assisted in the supervision of my work. Many thanks go to all the members of the Institute for Gravitational Research at Glasgow University (headed by Professor Jim Hough), and members of the Centre for Systems and Control, for the assistance and guidance they have provided.

Technical support has been supplied by Colin Craig, Angus McKellar, Alan Latta, Stephen Craig and Neil Robertson, as well as the staff of the Workshop (Gordon, Stuart, Ray and Davie).

This thesis was written using the most excellent L<sup>A</sup>T<sub>E</sub>X document preparation system.

# Preface

Chapter 1 contains a brief introduction to gravitational waves, including their production and means of detection. Noise sources that limit the sensitivity of ground based gravitational wave detectors are then presented. The subject matter in this chapter is derived from current literature and provides the motivation for the work contained within this thesis.

Chapter 2 provides an introduction to vibration isolation. In particular, the various vibration isolation elements, incorporated within a typical optic suspension, are given. These vibration isolation elements will subsequently be modelled using the bond graph methodology. The material in this chapter is drawn from published literature.

The bond graph methodology forms the basis for both modelling and control purposes and as such is detailed in Chapter 3. Again material in this section is derived from current literature.

In Chapter 4 a library of components, designed in the bond graph domain, for use in the modelling of multi-stage pendula, is presented. This library of components facilitates the construction of 2-dimensional multi-stage pendula models in a hierarchical fashion. To validate these components the mode frequencies of a two stage real physical pendulum are compared with an equivalent bond graph model. Furthermore, this chapter forms a tutorial for the construction and analysis of these bond graph pendulum models, including the use of a suitable (pre-existing) computer program designed to create and manipulate bond graphs.

Chapter 5 extends the basic wire model introduced in Chapter 4. Here components are created which permit the inclusion of bending dynamics and transverse modes of vibration.

The components detailed in Chapters 4 and 5, in conjunction with a new set of bond graph components, are used to create a Model Based Observer (MBO) controller for a double pendulum. This controller is detailed in Chapter 6 and is designed to damp the resonant modes of a double pendulum. This new controller has been implemented digitally and its performance compared with a previously designed analogue controller.

Chapter 7 details the investigation of a possible global control scheme which again is a model based observer design, incorporating a split feedback topology.

Finally, concluding remarks are made in Chapter 8.

# Summary

A prediction of Einstein's Theory of General Relativity is the creation of gravitational waves. These waves are created by an asymmetric acceleration of mass, travel at the speed of light, and produce a tidal strain in space. Unfortunately the interaction of a gravitational wave with matter is very weak, even violent astronomical events only produce strains of the order  $10^{-21}$  to  $10^{-22}$  at the Earth. So far, the only evidence of gravitational wave production has come from indirect sources. However, direct detection of the tidal strain is deemed possible with Earth based detectors. One such detector is based upon the Michelson interferometer.

GEO 600, a ground based Michelson interferometer (having 600 m arm lengths), is a joint project between Universities in the United Kingdom and Germany, which is expected to have the necessary sensitivity to detect gravitational waves. It is currently in the final stages of construction at a site near Hanover in Germany. This detector is designed to be sensitive to gravitational radiation down to a frequency of 50 Hz. This sensitivity limit ( $2 \times 10^{-22} \text{m}/\sqrt{\text{Hz}}$ ) is set by the thermal noise generated by the internal modes of the fused silica optics, which sets a minimum requirement for seismic isolation. The design goal for seismic isolation is to achieve a noise level a factor of 10 below that due to thermal noise at 50 Hz. To achieve this, various isolation elements are inserted between the ground and the sensitive optic. Chiefly, sensitive optics are suspended as multi-stage pendula. To facilitate the design and control of these suspension systems it is desirable to have accurate system models.

Here, it is believed that the bond graph methodology provides the ideal platform for the

modelling of these multi-stage pendula. This is because the methodology has the following properties; it is energy based, energy conserving, hierarchical, unambiguous, and equation, rather than, assignment based. With this in mind, this thesis presents a library of components with which simple (or complex) multi-stage pendula can be constructed in a modular fashion. This modularisation facilitates the easy interchange of components without having to re-model the complete system. To validate these bond graph components the mode frequencies of a real physical two stage pendulum are compared with those generated by the equivalent bond graph model.

Now, due to the resonant nature of pendula there is a requirement for active control of suspensions. Active control of these pendula takes two forms: ‘Local control’, so called because it acts on individual suspensions, is used to damp the resonant modes; whilst a ‘global control’ scheme is used to apply actuation forces to a multi-stage suspension such that the detector is maintained in the correct configuration.

The current local controller is an analogue implementation of a classically designed control law. In this thesis, having shown that this analogue implementation can be successfully implemented digitally, an alternative local controller design is presented. This is a Model Based Observer (MBO) design, based in the bond graph domain, and is a physical approach to controller design and as such is a more intuitive design methodology. Also, since it is based in the bond graph domain it provides a unified approach to system modelling and controller design. A digital realisation of a MBO designed local controller has been successfully implemented on a real physical system, and the comparison between the original analogue and new MBO controller is presented in this thesis.

Having shown that a controller design, using the MBO methodology, can suitably damp a real physical double pendulum the methodology is extended to produce, and investigate the properties of, a MBO designed global controller. This is a split feedback arrangement, designed to reduce actuation forces at the sensitive optic so that less powerful, but quieter, actuators may be used at the sensitive optic. This is achieved by applying lower frequency/greater displacement actuation higher up the chain of pendula where noisier more powerful actuation can be used. Although this controller is not implemented on a real phys-



ical system its potential is suitably demonstrated.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Preface</b>	<b>ii</b>
<b>Summary</b>	<b>iv</b>
<b>1 An Introduction to Gravitational Waves and their Detection</b>	<b>1</b>
1.1 An Introduction to Gravitational Waves . . . . .	1
1.2 Gravitational Wave Sources . . . . .	3
1.2.1 Burst . . . . .	3
1.2.2 “Chirp” . . . . .	4
1.2.3 Continuous . . . . .	5
1.2.4 Stochastic . . . . .	5
1.3 Gravitational Wave Detectors . . . . .	6
1.3.1 Introduction . . . . .	6
1.3.2 Resonant Mass Detectors . . . . .	6
1.3.3 Ground Based Interferometric Detectors . . . . .	7

1.3.4	GEO 600 . . . . .	9
1.4	Noise Sources in Ground Based Interferometric Gravitational Wave Detectors	11
1.4.1	Shot Noise . . . . .	11
1.4.2	Thermal Noise . . . . .	11
1.4.3	Acoustic and Seismic Noise . . . . .	12
1.4.4	Other Noise Sources . . . . .	13
1.4.5	Combining the Noise Sources: Projected Noise Performance of GEO 600 Detector . . . . .	13
<b>2</b>	<b>Vibration Isolation and GEO 600</b>	<b>15</b>
2.1	Introduction . . . . .	15
2.2	Vibration Isolation Requirements for GEO 600 . . . . .	15
2.3	The Mechanics of Passive Vibration Isolation . . . . .	16
2.3.1	Viscous and Structural Damping Forces . . . . .	19
2.3.2	The Characteristics of a Damped Second Order system . . . . .	19
2.4	GEO 600 and Passive Isolation . . . . .	21
2.4.1	Passive Isolation stacks . . . . .	21
2.4.2	Increased Vertical Isolation using Cantilever Springs . . . . .	22
2.4.3	Pendulum Suspension . . . . .	24
<b>3</b>	<b>Bond Graphs</b>	<b>26</b>
3.1	Introduction . . . . .	26

3.2	Modelling Philosophy . . . . .	26
3.3	An Overview of Bond Graphs . . . . .	27
3.4	Elementary Bond Graph Components . . . . .	28
3.4.1	One Port Components . . . . .	29
3.5	Structural Elements . . . . .	32
3.5.1	Junctions . . . . .	33
3.5.2	Coupling Elements . . . . .	34
3.6	Causality . . . . .	36
3.6.1	Junction Structures . . . . .	39
3.7	An Example to Illustrate Bond Graph Concepts . . . . .	41
3.7.1	The need for a INTF component . . . . .	41
3.7.2	INTF bond graph design . . . . .	42
3.8	Transformations and Representations . . . . .	44
3.9	Bond Graphs and Linearisation . . . . .	45
3.10	Conclusions . . . . .	47
<b>4</b>	<b>2-Dimensional Bond Graph Pendulum Models</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.2	The Model . . . . .	50
4.3	A Double Pendulum Bond Graph . . . . .	50
4.4	Support . . . . .	54

4.5	A Bond Graph of a Two Dimensional Mass . . . . .	56
4.5.1	The <b>UpperMass</b> Bond Graph Component . . . . .	57
4.5.2	The Bond Graph Component <b>UpperMassSusp</b> . . . . .	59
4.5.3	The Bond Graph Component <b>CoordTrans</b> . . . . .	61
4.6	Construction of a Wire Bond Graph . . . . .	69
4.6.1	Modeling a Semi-Rigid <b>Wire</b> Component . . . . .	69
4.6.2	The bond graph component <b>LinExt</b> . . . . .	71
4.6.3	Using MTT to check the consistency of this bond graph <b>wire</b> component . . . . .	73
4.6.4	The bond graph component <b>TwoWires</b> . . . . .	74
4.7	MTT and the Double Pendulum . . . . .	75
4.8	Comparing Double Pendulum Models with the Physical System . . . . .	81
4.9	Conclusions . . . . .	81
<b>5</b>	<b>The Bond Graph Modelling of Suspensions</b>	<b>84</b>
5.1	Introduction . . . . .	84
5.2	An Introduction to Modelling Wires . . . . .	85
5.2.1	Restoring Force Due To Wire Extension . . . . .	85
5.2.2	Transverse Motion of a Wire Under Tension . . . . .	86
5.3	The Creation of a Basic Wire Framework . . . . .	86
5.3.1	Modelling a Simple Pendulum As a Rigid Rod . . . . .	87
5.3.2	A Simple Light Rod . . . . .	87

5.4	Introduction to Modelling the Bending Dynamics of a Wire . . . . .	90
5.4.1	Bending Energy of a Flexible Wire . . . . .	91
5.4.2	The Bond Graph Model of a Bending Wire . . . . .	93
5.5	Modelling Transverse Modes of a Tensioned Wire . . . . .	96
5.5.1	Finite Modal Method of Modelling Transverse Modes . . . . .	97
5.5.2	A Lumped Method of Modelling Transverse Modes . . . . .	99
5.5.3	Mathematical foundation . . . . .	100
5.5.4	Bond Graph Synthesis . . . . .	100
5.5.5	Comparison of <b>Lumped</b> model with physical system . . . . .	105
5.6	Conclusions . . . . .	106
<b>6</b>	<b>Local Control</b>	<b>108</b>
6.1	Introduction . . . . .	108
6.2	Experimental Setup . . . . .	109
6.2.1	Sensors and Actuators . . . . .	110
6.3	The Existing Analogue Local Controller . . . . .	113
6.4	Digital Implementation of the Existing Analogue Control . . . . .	114
6.4.1	Introduction . . . . .	114
6.4.2	Digital Design in the Continuous Domain . . . . .	115
6.4.3	The Discrete Transfer Functions . . . . .	115
6.4.4	Creating a Digital Transfer Function From a Continuous Transfer Function Using Numerical Integration . . . . .	116

6.5	Model Based Observer (MBO) Control . . . . .	123
6.5.1	Mass-Spring Model Based Controller . . . . .	124
6.6	Local Control Design Using Bond Graphs . . . . .	129
6.7	Results . . . . .	133
6.8	Conclusions . . . . .	134
<b>7</b>	<b>Global Control</b>	<b>137</b>
7.1	Feedback to the Test Mass . . . . .	138
7.2	Split Feedback Control . . . . .	139
7.3	Split Feedback Controller Design Using A Bond Graph Model Based Ob- server (MBO) . . . . .	140
7.4	Split Feedback MBO Bond Graphs . . . . .	141
7.5	The Bond Graph Component <b>Actuator5</b> . . . . .	142
7.6	The Bond Graph Component <b>Compensator5</b> and Associated Subcomponents	143
7.6.1	The Bond Graph Component <b>Sys</b> . . . . .	144
7.6.2	The Bond Graph Component <b>Actuators5</b> . . . . .	144
7.6.3	The Bond Graph Component <b>VelContObs</b> . . . . .	144
7.6.4	The Bond Graph Component <b>PosnCont2</b> . . . . .	145
7.6.5	The Bond Graph Components <b>DcContObs4</b> , <b>DcContObs2</b> and <b>Dc- ContObs</b> . . . . .	147
7.6.6	Augmentation of the Observer . . . . .	149
7.7	Analysis of the Model Based Observer Split Feedback Controller . . . . .	150

7.8	Conclusions And Comments . . . . .	152
<b>8</b>	<b>Conclusions</b>	<b>158</b>
8.1	Future work . . . . .	159
<b>A</b>	<b>Transforming the Bond Graph Representation of a Mass–Spring System into an Equivalent Transfer Function Representation</b>	<b>160</b>
<b>B</b>	<b>Introduction to Bicausal Bond Graphs</b>	<b>163</b>
B.1	Bicausal Bond Graphs . . . . .	164
B.1.1	Bicausal Bonds . . . . .	164
B.1.2	Bicausal Junctions . . . . .	165
B.1.3	Source Sensor SS Components . . . . .	166
B.1.4	Flow and Effort Amplifiers . . . . .	166
B.2	Using Bicausal bond graphs to model the inverse dynamics of a sensor . . .	167
<b>C</b>	<b>Split feedback Numerical Parameters</b>	<b>170</b>



# List of Figures

1.1	<i>The effect of a gravitational wave on a ring of test particles . . . . .</i>	2
1.2	<i>A Schematic of a Michelson interferometer . . . . .</i>	7
1.3	<i>Schematic of a Michelson Interferometer incorporating Fabry-Perot cavities in each arm. . . . .</i>	9
1.4	<i>Schematic of the GEO 600 detector incorporating delay lines in each arm, and mirrors for power and signal recycling. . . . .</i>	9
1.5	<i>Predicted sensitivity of GEO 600 [1] . . . . .</i>	14
2.1	<i>Typical seismic readings for the GEO 600 site in Ruthe. For modelling purposes this is modelled as <math>10^{-7} / f^2 \text{ m} / \sqrt{\text{Hz}}</math> . . . . .</i>	16
2.2	<i>Schematic and bond graph of a Mass-Spring isolation system . . . . .</i>	17
2.3	<i>The transfer function of an undamped second order system . . . . .</i>	18
2.4	<i>The transfer functions of two 2nd order systems, one with infinite <math>Q</math> the other with a <math>Q</math> of 10 . . . . .</i>	20
2.5	<i>Schematic of the GEO 600 gravitational wave detector [2] . . . . .</i>	21
2.6	<i>Schematic of the GEO 600 main suspension . . . . .</i>	22
2.7	<i>Schematic of a cantilever spring . . . . .</i>	24

3.1	<i>A simple bond</i>	28
3.2	<i>The set of one port bond graph components: The source sensor <b>SS</b>, energy dissipator <b>R</b> and the two energy storage ports <b>C</b> and <b>I</b></i>	29
3.3	<i>The <b>SS</b> Component: effort and flow sources</i>	30
3.4	<i>The <b>0</b> and <b>1</b> bond graph junctions</i>	33
3.5	<i>Bond graphs for the transformer and gyrator</i>	35
3.6	<i>Causality for the one port components</i>	36
3.7	<i>An example of a simple junction structure [3]</i>	39
3.8	<i>An example of a multiport transformer</i>	40
3.9	<i>The <b>INTF</b> component: Integrates the flow variable</i>	42
4.1	<i>Double pendulum schematic and bond graph</i>	51
4.2	<i>The <b>SS</b> bond graph component and the use of named ports, port aliasing and vector bonds</i>	52
4.3	<i>Bond graph of an idealised pendulum support</i>	55
4.4	<i>The correspondence between the number of suspension points associated with a physical three dimensional mass and its two dimensional model</i>	57
4.5	<i>The bond graph component <b>UpperMass</b></i>	58
4.6	<i>The bond graph component <b>UpperMassSusp</b></i>	59
4.7	<i>The equivalent structure of a vector <b>I</b> junction as used within component <b>UpperMassSusp</b></i>	60
4.8	<i>Definition of coordinate system</i>	61

4.9	<i>Transformation relating suspension point velocities to centre of mass velocities.</i>	63
4.10	<i>Geometry of a wire</i>	70
4.11	<i>Bond Graph of a wire element</i>	71
4.12	<i>Bond graph element that provides restoring forces due to linear extension</i>	71
4.13	<i>The bond graph component <b>TwoWires</b></i>	75
5.1	<i>The bond graph of a modified two port mass component used to model a light rod</i>	88
5.2	<i><b>Hinge</b>, <b>HingeSpring</b> and <b>HingedLightRod</b> bond graph components. <b>HingedLightRod</b> utilises <b>LightRod</b> (Figure 5.1) and the two types of hinge component to produce a “simple wire” component.</i>	89
5.3	<i>A bond graph model of a simple pendulum with two degrees of freedom. Key components are <b>SpringLightRod</b> and a single port mass element.</i>	90
5.4	<i>The basic framework for the modelling of wire dynamics.</i>	91
5.5	<i>General boundary conditions for bending of a stiff beam under tension</i>	92
5.6	<i>A four port bond graph C component</i>	94
5.7	<i>Effective boundary conditions for an embedded <b>Bending</b> component</i>	95
5.8	<i>A 3 port bond graph component combining extension and bending characteristics of a flexible wire</i>	95
5.9	<i>The inclusion of bending dynamics into the basic wire model. Note that the only difference between this and the component <b>LinExtWire</b>.abg.fig is that <b>LinExt:wire</b> has been replaced by <b>Bending:wire</b></i>	96

5.10	<i>The bond graph of the modal method of modelling the modes of a continuous system. Here four modes are to be model with two force inputs. . . . .</i>	97
5.11	<i>Force diagram for a short segmented massive wire undergoing transverse vibrations . . . . .</i>	99
5.12	<i>A schematic and bond graph representation of the transverse motion of a wire under tension . . . . .</i>	102
5.13	<i>The implementation of the lumped model within MTT . . . . .</i>	103
5.14	<i>Hierarchical bond graph of the component <b>Wire</b> incorporating the <b>Lumped</b> representation of transverse modes . . . . .</i>	106
6.1	<i>A photograph of the double pendulum system . . . . .</i>	109
6.2	<i>Schematic of physical double pendulum . . . . .</i>	110
6.3	<i>A schematic of the sensor actuator combination . . . . .</i>	111
6.4	<i>Variation of force with respect to coil magnet separation . . . . .</i>	112
6.5	<i>Theoretical bode plot of analogue local controller . . . . .</i>	113
6.6	<i>A block diagram representation of a sampled data system . . . . .</i>	114
6.7	<i>Mappings of the left half <math>s</math>-plane (stable region) to the <math>z</math>-plane via the rules of Table 6.1. Stable <math>s</math>-plane poles map into the shaded regions in the <math>z</math>-plane. The unit circle is shown for reference. . . . .</i>	117
6.8	<i>A graphical representation of the direct form of a second order digital transfer function . . . . .</i>	120
6.9	<i>Comparison of local control implementations . . . . .</i>	122
6.10	<i>Schematic of a model based controller . . . . .</i>	123
6.11	<i>Bond graph and schematic of a mass spring system . . . . .</i>	125

6.12	<i>Hierarchical bond graph of a model based controller</i> . . . . .	125
6.13	<i>Bond graph containing the model based observer and inverse sensor model (see Figure 6.10)</i> . . . . .	126
6.14	<i>The response of the “system” and model to a step input at the setpoint</i> . . .	129
6.15	<i>The response of the “system” and model to a impulsive disturbance of the mass</i> . . . . .	130
6.16	<i>Top level MBO bond graph controller</i> . . . . .	130
6.17	<i>Bond graphs - Actuators2 and associated subcomponent Xact.</i> . . . .	131
6.18	<i>Bond graphs - Compensator2 and subcomponent ContObs</i> . . . . .	132
6.19	<i>System response to impulse at intermediate mass - channel 3 (Top -&gt; orig- inal, Bottom -&gt; MBO)</i> . . . . .	135
6.20	<i>System response to impulse at intermediate mass - channel 4 (Top -&gt; orig- inal, Bottom -&gt; MBO)</i> . . . . .	136
7.1	<i>One possible electrostatic drive configuration</i> . . . . .	138
7.2	<i>Schematic of split feedback topology</i> . . . . .	139
7.3	<i>Schematic of a GEO 600 gravitational wave detector suspension. Notice that this is a three stage pendulum with two stages of cantilever blades.</i> . .	140
7.4	<i>Top level hierarchical bond graph of a model based observer as used to design a split feedback global controller</i> . . . . .	141
7.5	<i>Bond Graph Component Actuators5: Defines the location of the actuators, two at the upper mass, two at the intermediate mass, and one at the test mass.</i>	142
7.6	<i>Bond graph component Compensator5: Used to produce a split feedback compensator.</i> . . . . .	143

7.7	<i>The bond graph of the component <b>VelContObs</b>. It extends the <b>ContObs</b> component by providing an extra port to convey the flow variable at the 1 junction 1:s to external components</i>	144
7.8	<i>The <b>PosnCont2</b> component</i>	145
7.9	<i>Bond graph component <b>PosnCont</b>, provides proportional and integral action for position control</i>	146
7.10	<i>The bond graph component <b>DcContObs</b> provides an extra port c.f. <b>ContObs</b> for the inclusion of integral action forces associated with position control.</i>	147
7.11	<i>The bond graph component <b>DcContObs2</b> containing two instances of the component <b>DcContObs</b></i>	148
7.12	<i>The bond graph component <b>DcContObs4</b> containing two instances of the component <b>DcContObs2</b></i>	148
7.13	<i>A modified <b>DcContObs</b> component incorporating integral observer action, which is used to drive the displacement error between system and model to zero</i>	149
7.14	<i>The actuation forces at the upper, intermediate and test masses in response to a step input at the test mass setpoint</i>	153
7.15	<i>The resultant displacement of the upper, intermediate and test masses to a step input at the test mass</i>	154
7.16	<i>The impulse response of the upper, intermediate and test masses - impulse applied at upper mass</i>	155
7.17	<i>The bode plots: Input is position setpoint (test-mass) and outputs are actuator forces to masses: Magenta:- test mass, Other colours are those to the upper and intermediate masses. These all have approximately the same magnitude because the the coefficients <math>c_u</math> and <math>c_i</math> have been assigned same values.</i>	156

7.18	<i>Bond graph of a GEO 600 gravitational wave detector suspension. Notice that this is a three stage pendulum with two stages of cantilever blades.</i>	157
A.1	The schematic and causal bond graph of a Mass–Spring System	161
B.1	<i>Hierarchical bond graph component utilising bicausal bonds to model the inverse dynamics of inverse sensor dynamics</i>	164
B.2	<i>Bicausal bonds</i>	165
B.3	<i>The propagation of bicausality at a junction</i>	165
B.4	<i>Bicausality and source sensor components SS</i>	167
B.5	<i>Possible bicausal assignments as applied to flow and effort amplifiers</i>	168
B.6	<i>The annotated bond graph of the bond graph component modelling inverse sensor dynamics</i>	169

# List of Tables

3.1	<i>Effort and flow variables by domain . . . . .</i>	29
3.2	<i>Bond Graph capacitive component (C) modelling a spring. i.e. a store of potential energy. . . . .</i>	32
3.3	<i>A Bond Graph inductive component (I) modelling a simple mass element. i.e. a store of kinetic energy. . . . .</i>	33
3.4	<i>Integral and differential causal relationships for the C and I components . .</i>	37
3.5	<i>Analysis of the compound INTF component . . . . .</i>	43
3.6	<i>A set of ordered equations for the INTF component . . . . .</i>	44
4.1	<i>Double_lbl.txt: Double pendulum label file . . . . .</i>	53
4.2	<i>Form of the information fields for a top level bond graph . . . . .</i>	53
4.3	<i>Support_lbl.txt: Support label file . . . . .</i>	54
4.4	<i>Table of possible SS field arguments . . . . .</i>	56
4.5	<i>The structure files for the CoordTrans component . . . . .</i>	65
4.6	<i>UpperMass_lbl.txt file . . . . .</i>	67
4.7	<i>LinExt_lbl.txt File . . . . .</i>	72



4.8	<i>The structure file for the Wire bond graph. Inputs are flows and outputs are efforts . . . . .</i>	73
4.9	<i>TwoWires.lbl.txt file . . . . .</i>	76
4.10	<i>The structure table for the double pendulum hierarchical bond graph . . . .</i>	78
4.11	<i>Double.sspar.r file: The steady state parameter file for system Double . . .</i>	79
4.12	<i>Table of eigenvalues for prototype double pendulum . . . . .</i>	82
4.13	<i>Double.numpar.m: The numerical parameter file for system Double . . . .</i>	83
5.1	<i>The label file for component Lumped . . . . .</i>	104
5.2	<i>Comparison of mode frequencies: Experiment Vs 1-d Wave Equation Vs Lumped Method . . . . .</i>	105
6.1	<i>s substitutions (in z) corresponding to numerical integration. . . . .</i>	116
6.2	<i>A fragment of computer code for the implementation of a second order digital transfer function represented by the direct form [4] . . . . .</i>	119
6.3	<i>The structure files for the mass spring velocity damping model based observer</i>	127
7.1	<i>A fragment of the numpar file (numerical parameter file) for the MBO split feedback controller . . . . .</i>	151
C.1	<i>Numerical parameters used in global control design . . . . .</i>	170
C.2	<i>Numerical parameters used in global control design continued . . . . .</i>	171
C.3	<i>Numerical parameters as used in global control designed continued . . . .</i>	172

## **Chapter 1**

# **An Introduction to Gravitational Waves and their Detection**

This chapter introduces the concept of gravitational waves, why we may wish to detect them, their effects on matter, and means of production. Various gravitational wave detectors, with which detection of gravitational waves are deemed probable, are then introduced. Finally noise sources that limit the detector sensitivity are discussed.

### **1.1 An Introduction to Gravitational Waves**

Why do we aim to detect gravitational waves? Firstly, we wish to verify, or otherwise, relativistic gravitational theories. Secondly, gravitational waves and their detection will provide us with new insights into the workings of the universe. Today's telescopes gather information from electromagnetic radiation which interacts with, and is scattered by, matter. However gravitational waves pass through matter with impunity, thus revealing the internal dynamics of stars. Further, as a result of having to manufacture detectors of very high sensitivity, gravitational wave research groups are pushing back the frontiers of material science, seismic vibration isolation, laser technology and optics.

The existence of gravitational waves were predicted by Einstein's General Theory of Relativity (1916) [5]. They are analogous to electromagnetic waves in that they travel at the speed of light and are transverse waves. However, whereas electromagnetic waves are created by the acceleration of charge, gravitational waves are produced by the acceleration of mass and since mass has only one "sign" these accelerations need to be asymmetric. It should also be appreciated that the gravitational force is extremely weak compared with the electrostatic: The ratio of gravitational to electrostatic force for a pair of protons is  $\approx 10^{-36}$ . Hence extremely large masses undergoing great accelerations are required to produce detectable waves. For this we turn to violent astronomical events such as supernovae and black hole formation and interaction, (see Section 1.2).

The effect of a gravitational wave is to produce a tidal force. As the wave propagates it produces a strain in space which is characterised by the dimensionless amplitude  $h$ , defined as

$$h = \frac{2\Delta L}{L} \quad (1.1)$$

Where the length  $L < \lambda_{gw}$ . The effect a wave has on a ring of test particles can be seen in Figure 1.1. Here the wave is propagating normal to the page and demonstrates the "+" (plus) polarisation. There is one other independent polarisation state known as the "x" (cross) and has the same effect but rotated by 45 degrees about the propagation axis.

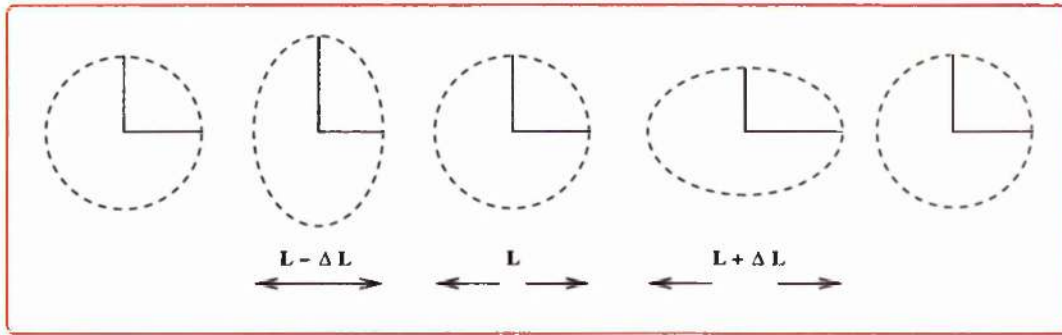


Figure 1.1: *The effect of a gravitational wave on a ring of test particles*

## 1.2 Gravitational Wave Sources

Ground based detectors are not expected to have the ability to detect gravitational waves below a frequency of 1 Hz. This is due to Earth atmospheric density fluctuations and Earth vibrations causing perturbations of the gravitational field. Sources are predicted to have a maximum frequency of  $\sim 10^4$  Hz. This upper limit arises from the minimum mass (typically  $1 M_{\odot} = 1$  solar mass) needed to create a compact collapsed star massive enough to strongly emit gravitational waves. The frequency of the gravitational wave is determined by the time taken for the wave to traverse the radius of the star. Gravitational wave sources can be categorised into the following types; burst, “chirp”, continuous and stochastic, and are now discussed.

### 1.2.1 Burst

Bursts of gravitational waves are believed to be created by catastrophic explosions of stars. These explosions are called supernovae of which there are two types. Type I supernovae are generated when a white dwarf star, accreting material, possibly from a binary companion, undergoes a thermonuclear explosion. This occurs when it's mass exceeds the critical mass (Chandrasekhar mass  $M_{\text{ch}} = 1.4M_{\odot} \approx 3 \times 10^{30} \text{ Kg}$ ).

The catastrophic end to the life of a massive star, mass  $> 8M_{\odot}$ , results in type II supernovae. When a massive star has burnt all its nuclear fuel the radiation pressure can no longer balance the gravitational force and the core collapses. Thus the density of the core increases until the star's central density is close to that of an atomic nucleus. Matter that subsequently falls onto this central core rebounds, producing a shock wave that travels to the surface of the star, ejecting the outer layers. If the central core is not spherical when this process occurs, gravitational waves will be produced. Depending on the initial stellar mass, the remnant star will be a neutron star (possibly a pulsar) or a black hole (mass  $> 10M_{\odot}$ ). Such events, out to the Virgo cluster (15Mpc), are estimated to occur at a rate of 15 per month [6].

The expected strain amplitude produced from a burst source, a distance  $r$  from the Earth,

with frequency  $f$ , and emitting gravitational wave energy  $E$ , is given by Equation 1.2 [7].

$$h \sim 5 \times 10^{-22} \left[ \frac{E}{10^{-3} M_{\odot} c^2} \right]^{\frac{1}{2}} \left[ \frac{1 \text{ kHz}}{f} \right] \left[ \frac{\tau}{1 \text{ ms}} \right]^{-\frac{1}{2}} \left[ \frac{15 \text{ Mpc}}{r} \right] \quad (1.2)$$

This is scaled, based upon the conversion of  $\approx 0.1\%$  of the available energy into gravitational radiation [8].

### 1.2.2 “Chirp”

Coalescence of compact binary systems, of either neutron stars or black holes, produces a unique gravitational wave form known as “chirp”. The process, of two massive objects inspiralling about their common mass centre, produces gravitational waves. As the separation of the binary decreases, the frequency and amplitude of the radiation increases. Eventually the binary reaches its final stable orbit and a subsequent “plunge” phase, with an expected high emission of gravitational wave radiation, occurs. Typically the inspiral stage is expected to last  $10^8 - 10^{10}$  years, whilst the plunge phase will only last  $\sim 15$  minutes, and in this final stage the gravitational wave’s frequency will rise from  $\sim 10$  Hz to  $\sim 1$  kHz. The Sensitivity of GEO 600 (see Section 1.4.3) will restrict detection to the plunge stage of this phenomenon.

The strain amplitude associated with a binary coalescence is given by [7]:

$$h \approx 2.6 \times 10^{-23} \left( \frac{\mathcal{M}}{M_{\odot}} \right)^{\frac{5}{3}} \left( \frac{f}{100 \text{ Hz}} \right)^{\frac{2}{3}} \left( \frac{100 \text{ Mpc}}{r} \right) \quad (1.3)$$

Where  $f$  is the gravitational wave frequency,  $r$  is the distance from the Earth and  $\mathcal{M}$  is the chirp mass defined as

$$\mathcal{M} = \frac{(M_1 M_2)^{\frac{3}{5}}}{(M_1 + M_2)^{\frac{1}{2}}} \quad (1.4)$$

$M_1$  and  $M_2$  are the masses of the binary. Thorne [9] estimates an event rate of 3 per year out to 200 Mpc

Radio observations of the binary pulsar PSR 1913 + 16 [10] provide evidence of this decrease in periodicity. The observed rate of inspiral of PSR 1913+16 is accurate to within 1%

(experimental accuracy) of the theoretical value predicted by General Relativity. This first indirect observation of gravitational radiation resulted in Russell Hulse and Joseph Taylor being awarded the 1993 Nobel Prize.

### 1.2.3 Continuous

Single rotating neutron stars with an asymmetric distribution of mass are expected to radiate gravitational waves. For such monochromatic sources it is possible to increase the sensitivity of the detector by a factor of  $\sqrt{\tau_{int}}$  where  $\tau_{int}$  is the integration time. Further, since the received signal will be modulated, both in amplitude and frequency, by the Earth's rotation, it should be possible to verify the detection of a source with a single detector.

### 1.2.4 Stochastic

Similar to the background cosmic microwave radiation, it is expected that a stochastic background gravitational wave radiation exists. This background radiation appears in a single detector as noise. There are two ways of detecting it:

- If the unwanted noise sources were well characterised, and if the observed noise were greater, then any excess noise could be attributed to the gravitational wave. It is unlikely that ground detectors will be sensitive enough to distinguish this, although it is possible that a space detector could [8].
- With two detectors it may be possible to cross-correlate their output. By multiplying the outputs, the instrument noise, assumed independent, will cancel, whilst the gravitational wave noise will sum systematically. This assumes that the two detectors are close enough together that any given component of the gravitational wave noise is correlated, i.e. the separation must be less than the wavelength of the gravitational wave [8].

## 1.3 Gravitational Wave Detectors

### 1.3.1 Introduction

Detectors of very high sensitivity are required to detect gravitational waves. Two main detector designs, predicted to have the necessary sensitivity, have been proposed. The first of these, the resonant bar detector, was proposed and developed by Weber [11]. The second is the interferometric type of detector of which both ground and space borne detectors are being developed.

### 1.3.2 Resonant Mass Detectors

Resonant mass detectors generally operate narrow band at 1 kHz with a bandwidth of 1 Hz and at low temperature ( $\sim 4$  K), having a sensitivity  $h \approx 6 \times 10^{-19}$  [12] [13] [14] [15]. Essentially they are massive (a few tonnes), high quality factor (Q), right cylindrical bars. High Q materials have the advantage that once excited they have a long ring down period. This increases the sensitivity of the detector through increased integration time. The passage of a suitably orientated gravitational wave will produce a strain in the detector, possibly causing a detectable motion of the bar's ends at its fundamental modal frequency. Associated with the measurement of this motion is sensor and thermal noise. An integration time that balances the conflict between a short integration time/higher sensor noise (high bandwidth) and longer integration time/increased thermal noise, is needed. Thermal noise can be reduced by using a very high Q material cooled to liquid helium temperatures (4.K). Bar detectors are currently under development at Stanford, Louisiana state university (ALLEGRO), Rome (NAUTILUS), CERN (EXPLORER) and Australia (NIOBE).

Increased bar detector sensitivity can be achieved by having more antennas and by increasing their mass. Extra mass also reduces thermal noise. A spherical resonant mass detector achieves both of these requirements. It has 15 times as much mass, as a bar of the same fundamental frequency, and has 5 independent quadrupole moments, compared with the bar's single mode. A gravitational wave passing through a spherical detector will excite these

modes. The direction and polarisation of the gravitational wave can be determined from the ratio of the mode amplitudes [16]. There are proposals to build spherical detectors in the USA (TIGA)[17] and the Netherlands (GRAIL)[18].

### 1.3.3 Ground Based Interferometric Detectors

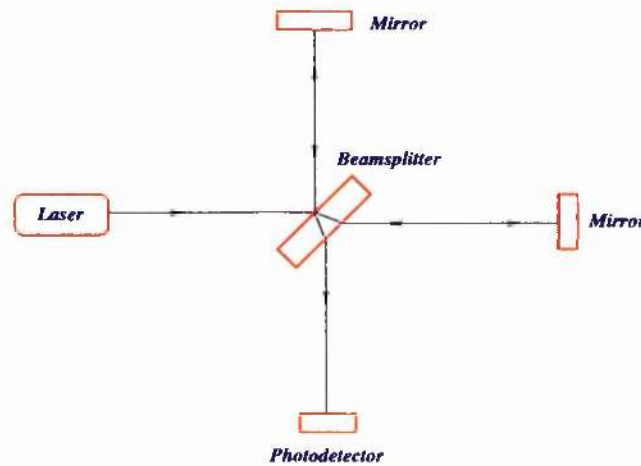


Figure 1.2: A Schematic of a Michelson interferometer

Ground based interferometric detectors are inherently broadband and are thus ideal for observing signals that are either short burst or have frequencies that change with time. It is expected that GEO 600 will have a bandwidth from 50 Hz to several kHz. These detectors are based on the Michelson Interferometer.

The Michelson Interferometer is formed by a beamsplitter and two mirrors. Coherent light, split into two beams by the beam splitter, is incident on the mirrors as shown in Figure 1.2. Recombination of the light, at the beamsplitter, results in an interference pattern, the form of which depends on the phase difference between the two beams. This phase difference is introduced through the interferometer having unequal arm lengths. When in operation the detector is configured to have a null output; that is, it is maintained on a dark fringe. A gravitational wave passing through the detector causes the relative arm length to change, thus phase modulating the laser light. The effect of this modulation is to produce sidebands about the laser light frequency (carrier frequency). The gravitational wave information is



contained within these sidebands. It is this signal, by feedback to one or more of the test masses, that maintains the detector on a dark fringe.

The mirrors of the detector must be free to move under the influence of a gravitational wave without being influenced by seismic and acoustic noise. Hence mirrors and other optics are suspended as pendula and the detector is operated under very high vacuum (typically  $10^{-8}$  mbar). As can be seen in Figure 1.1 (Page 2) an interferometric detector is ideally suited to detecting the quadrupole nature of gravitational waves. Further, the sensitivity of a detector is enhanced by increasing its size: The optimum having arm lengths  $L = \lambda_{gw}/4$ . For a gravitational wave of 1 kHz the optimal length is 75 km which, due to practical considerations such as cost, is too long for ground base detectors. However a detector's effective arm length can be increased by incorporating either delay line or Fabry-Perot cavities into each arm.

The Glasgow prototype detector is of the Fabry-Perot kind. A schematic is shown in Figure 1.3. Here, a resonant cavity is created in each arm through the use of partially transmitting mirrors. Photons entering the cavity, via the partially transmitting mirror, travel back and forth between the mirrors. This happens many times before the photons leave the cavity via the partially transmitting mirror. Thus the power in the cavity is much greater than the input power and the effective storage time of the light is increased.

GEO 600 is an example of a delay line interferometric gravitational wave detector. A schematic is shown in Figure 1.4. Here the storage time is increased by reflecting light at the outboard mirror onto an inboard mirror situated near the beamsplitter: Doubling the detector's effective arm length.

The choice of the delay line, over the Fabry-Perot configuration, stems from the difficulties in implementing the Fabry-Perot cavity. Although a delay line interferometer has larger mirrors and may suffer from scattered light contaminating the beam, an interferometer with Fabry-Perot cavities may suffer from thermal lensing. This is due to the difficulty of producing partially transmitting materials of very low loss. Moreover, extremely accurate control is needed to keep the two cavities independently on resonance.

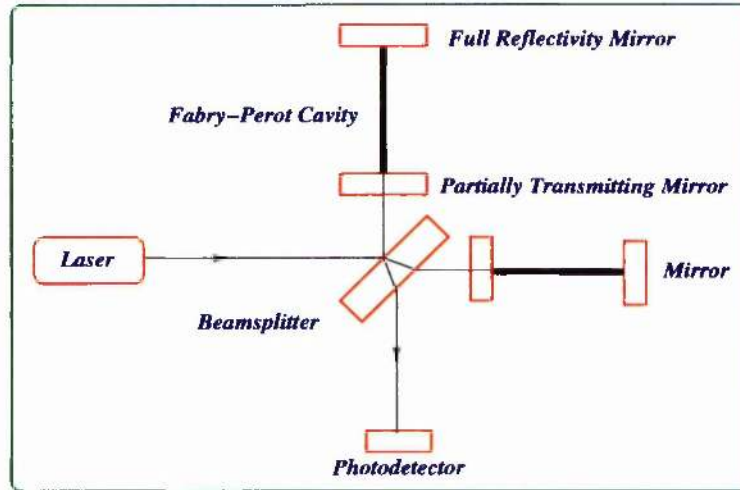


Figure 1.3: *Schematic of a Michelson Interferometer incorporating Fabry-Perot cavities in each arm.*

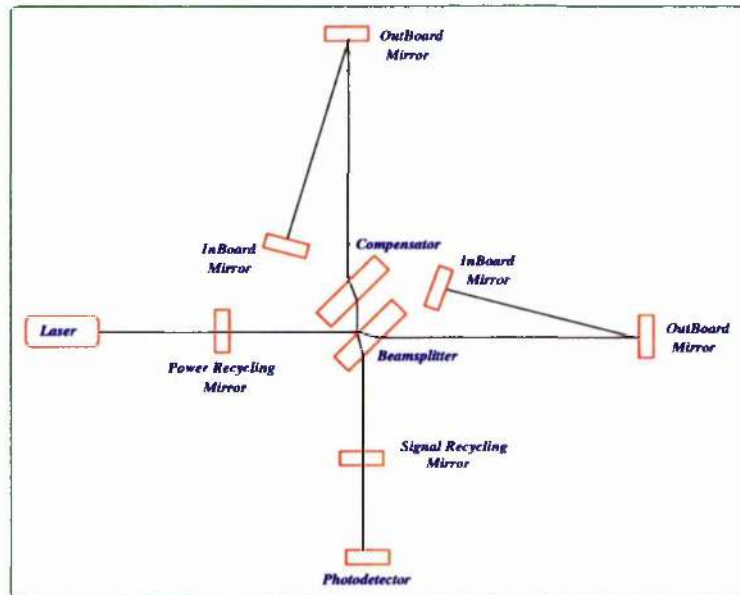


Figure 1.4: *Schematic of the GEO 600 detector incorporating delay lines in each arm, and mirrors for power and signal recycling.*

### 1.3.4 GEO 600

The GEO 600 gravitational wave detector is sited at Ruthe (a few kilometres outside Hanover) and is nearing the completion of its construction stage. The project is a joint

collaboration between universities in the United Kingdom (Glasgow and Cardiff) and universities in Germany (Max Plank Institute [Hanover] and Potsdam). The main features of GEO 600 are as follows:

- Each arm is 600 metres long and thus with the inclusion of a delay line the effective length is 1200 m.
- The compensator is added to provide the same optical path length in each arm.
- As stated earlier the detector is maintained on a dark fringe. As a result, light, after recombination at the beamsplitter, is lost to the detector. The power recycling mirror [19] is used to reflect this light back into the interferometer. This increases the power in the arms of the detector, thus reducing the effects of shot noise (see Section 1.4.1). In GEO 600 the power is expected to be increased by a factor of  $\sim 2000$ . To achieve this mirrors of very low loss are required.
- The signal recycling mirror [20] operates in a similar manner as the power recycling mirror; returning any recombined signal back into the detector. The position of this mirror determines the frequency of the resonant mode and its reflectivity determines the operational bandwidth.

In operation the output intensity of the detector is maintained at a null by control of the arm lengths. A gravitational wave will change the output intensity. Hence the feedback signal, to prevent a change in arm length, will contain the gravitational wave signal. The control scheme employed for this purpose is known as *global control*. Global control is discussed in Chapter 7. Obviously the output signal will contain any noise within the system and as such are discussed in the next section.

## 1.4 Noise Sources in Ground Based Interferometric Gravitational Wave Detectors

Various noise sources, such as seismic, thermal and shot noise, limit the performance of gravitational wave detectors. These are detailed below.

### 1.4.1 Shot Noise

A fundamental limit to a detector's sensitivity is the shot noise. The interferometric signal is directed onto a photodetector to produce a photocurrent. The quantised nature of light results in statistical fluctuations of this current. The spectral density fluctuation is given by the Schottky equation:

$$S_i(F) = 2ei \quad (1.5)$$

where  $e$  is the electronic charge and  $i$  is the photocurrent. The photocurrent is given by:

$$i = \frac{e\eta P}{\hbar\omega} = \frac{e\eta P\lambda}{2\pi\hbar c} \quad (1.6)$$

where  $\hbar$  is Planck's constant/ $2\pi$ ,  $c$  is the speed of light in a vacuum,  $P$  is the light power and  $\eta$  is the photodiode quantum efficiency.

The resulting displacement noise is :

$$\tilde{x} = \left( \frac{\hbar c}{\pi} \frac{\lambda}{\eta P} \right)^{1/2} \text{ m}/\sqrt{\text{Hz}} \quad (1.7)$$

### 1.4.2 Thermal Noise

Thermal motion occurs in all atoms at temperatures above absolute zero. Where the *equipartition theorem* associates  $\frac{1}{2}k_B T$  with each degree of freedom,  $T$  is the temperature and  $k_B$  is Boltzmann's constant. The random thermal motion of these atoms is manifest as a macroscopic displacement noise. Work by Callen et.al. [21][22] determined the relationship between the thermal fluctuations of a system and its dissipation (Fluctuation Dissipation theorem)

The pendulum modes of the system, transverse modes of the suspension wires and internal modes of the masses all contribute to the overall thermal noise signature. The thermal noise characteristics are illustrated with an harmonic oscillator.

The power spectral density for a harmonic oscillator is given by:

$$\tilde{x}^2(\omega) = \frac{4k_B T \omega_0^2 \phi(\omega)}{\omega m \left[ (\omega_0^2 - \omega^2)^2 + \omega_0^4 \phi(\omega)^2 \right]} \quad (1.8)$$

where,  $k_B$  is Boltzmann's constant,  $T$  is the temperature,  $\omega_0$  is the resonant frequency,  $m$  is the mass of the pendulum and  $\phi(\omega)$  is the loss tangent (defined below).

The loss tangent<sup>1</sup> is defined as the phase lag between the applied force and the corresponding displacement. A low loss tangent corresponds to a high Quality factor (Q). Where  $Q = \frac{1}{\phi(\omega)}$ . Hence, to minimise thermal noise, materials of very high Q are used. In GEO 600 the mirror and it's suspension are made of fused silica having a Q of a few  $10^6$ . When a pendulum swings most of its potential energy is stored in the gravitational field and not in the bend of the wire. As a result the Q of the fused silica suspension has an effective Q 100 times greater than that of the bulk material. A consequence of using high Q materials is that noise at the mode frequencies is resonantly enhanced. Hence, to damp resonantly enhanced motion due to seismic noise, pendulum modes are actively controlled. Also, the material and dimensions of the sensitive optic are chosen such that the internal modes (typically > 30 kHz) are outwith the operational bandwidth. Finally the violin modes are notch filtered out of the signal processing. The overall thermal noise, as contributed by the various modes of the test mass, are calculated to be  $7 \times 10^{-20} \text{m}/\sqrt{\text{Hz}}$  at 50 Hz [23]. This is based upon a measured loss tangent of  $2 \times 10^{-7}$  for the fused silica optic.

### 1.4.3 Acoustic and Seismic Noise

The suspended optics of GEO 600 are housed in a high vacuum. The high vacuum has the effect of reducing fluctuations in the refractive index and affords protection from acoustic noise.

---

<sup>1</sup> $\phi(\omega)$  may also be referred to as the loss angle or loss factor.

The seismic motion is approximately  $(10^{-7}/f^2) \text{ m}/\sqrt{\text{Hz}}$  from 1 Hz to  $\sim 100$  Hz. The true nature of this motion will depend on site location. The filtering of seismic motion, from ground to sensitive optic, is primarily achieved by hanging them as multi-stage pendula. Above their natural frequency each pendulum stage affords a  $(f_0^2/f^2)$  attenuation from seismic noise, where  $f_0$  is the pendulum's natural frequency. Additional attenuation is achieved from passive/active *isolation stacks*, (see Section 2.4.1). Layers of rubber and metal constitute the passive section, whilst the active section consists of piezoelectric actuators and feedback electronics. The effect of using mechanical systems to attenuate high frequency motion results in an enhancement of the seismic motion at the mechanical resonant frequencies. Hence these modes need to have low Q factors (high loss). For stacks lossy rubber is used whilst the pendulums are electronically controlled. This control scheme, since it acts on individual suspensions, is known as "local control" and will be discussed in Chapter 6.

#### 1.4.4 Other Noise Sources

Fluctuations in power, frequency and beam geometry of the input laser light are also sources of detector noise. The frequency and power are stabilised using feedback loops [20][24]. For methods of beam geometry stabilisation see [25][26]. Electronic, sensor and actuator noise also need to be minimised.

#### 1.4.5 Combining the Noise Sources: Projected Noise Performance of GEO 600 Detector

Figure 1.5 shows how the above noise sources contribute to the overall noise sensitivity of GEO 600, Where the meaning of the various coloured lines are given below:

- Green = Total noise signature
- Black = Shot noise (From a 5W,  $1.06\mu\text{m}$  and a recycling factor of 2000).
- Blue = Represents the thermal noise and is a combination of the thermal noise produced by the pendulum (assumes loss factor of  $1.4 \times 10^{-7}$ ) and the internal modes of



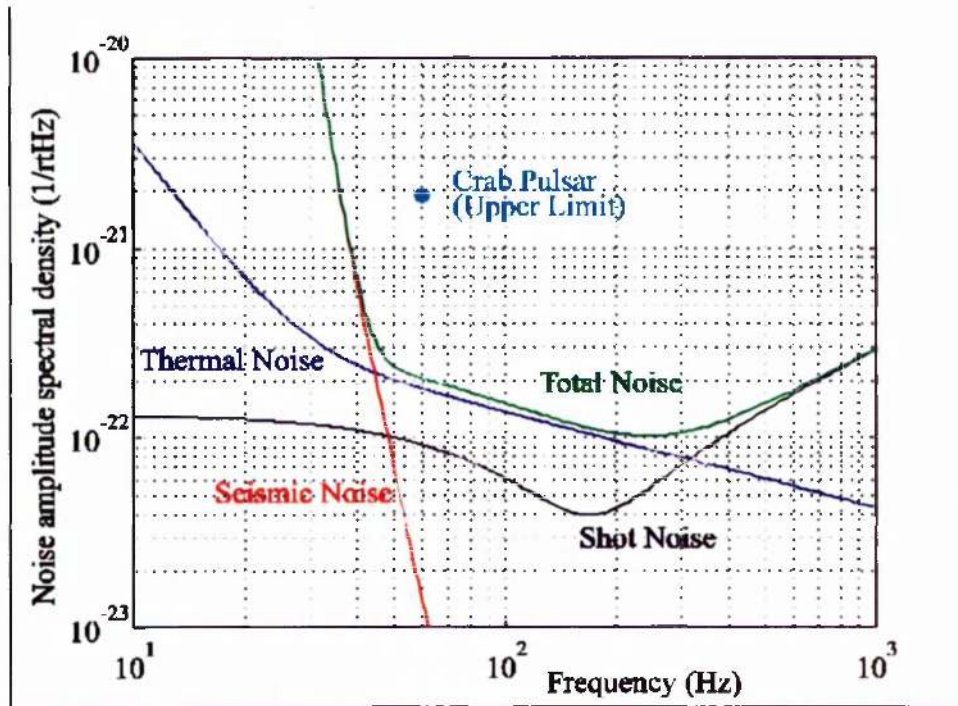


Figure 1.5: Predicted sensitivity of GEO 600 [1]

the sensitive optic.

- Red = Seismic noise.

Note that at higher frequencies the dominant noise source is shot noise. Thermal noise, associated with the internal modes of the silica test masses dominates between approximately 50 Hz and 200 Hz. The resultant test mass motion, due to thermal noise at 50 Hz ( $7 \times 10^{-20} \text{ m}/\sqrt{\text{Hz}}$ ), sets the design goal for seismic isolation. This states that the motion of a GEO 600 sensitive optic, due to seismic noise, should be a factor of 10 below that induced by thermal noise. As can be seen, the “seismic wall” dominates at lower frequencies. This results in an optimum strain sensitivity across a frequency band from  $\sim 50$  Hz to 1 kHz. The next chapter details the necessary vibration isolation mechanisms used to meet the design goal for seismic isolation.

## Chapter 2

# Vibration Isolation and GEO 600

### 2.1 Introduction

In this chapter elements of vibration isolation within the context of the GEO 600 project are presented. Firstly, the vibration isolation requirements for GEO 600 are detailed. The fundamentals of passive vibration isolation are then given, and finally the detailing of the various passive isolation components that constitute an optic suspension are detailed.

### 2.2 Vibration Isolation Requirements for GEO 600

Towards the end of the previous chapter (Section 1.4.5) it was stated that the seismic isolation requirement for a sensitive optic is: At 50 Hz the test mass motion, due to seismic noise, should be a factor of 10 below that due to thermal noise. At 50 Hz the thermal noise is expected to be  $2 \times 10^{-22} \text{m}/\sqrt{\text{Hz}}$  which, for the GEO 600 optical scheme, corresponds to a test motion of  $7 \times 10^{-20} \text{m}/\sqrt{\text{Hz}}$  [27]. The consequences of this are now discussed.

Having chosen a detector site it is necessary to characterise the seismic noise at that location. This was undertaken for the site at Ruthe (the GEO site) and a typical seismic profile is shown in Figure 2.1. Although the spectrum will vary, dependent on site and time, the



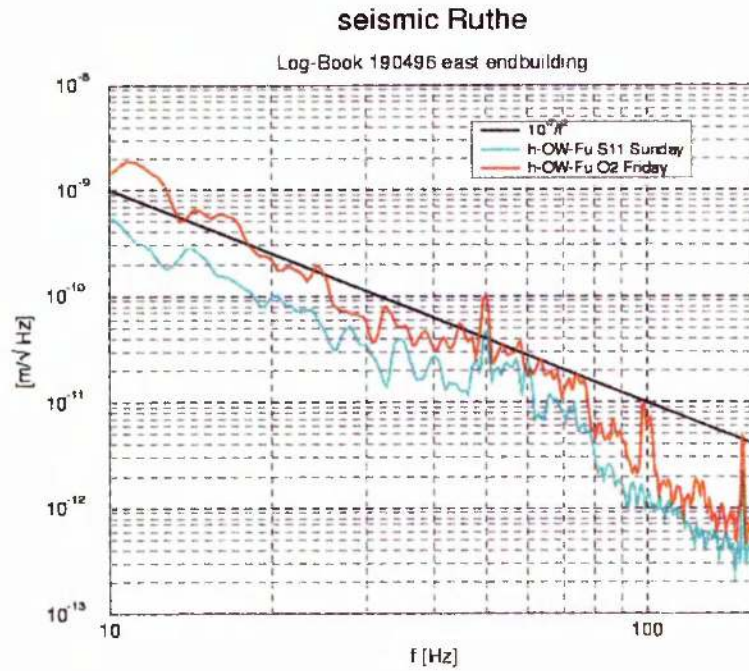


Figure 2.1: *Typical seismic readings for the GEO 600 site in Ruthe. For modelling purposes this is modelled as  $10^{-7}/f^2 \text{ m}/\sqrt{\text{Hz}}$*

following approximation has been adopted:

$$x(f) = x_0 \left( \frac{1}{f^2} \right) \frac{\text{m}}{\sqrt{\text{Hz}}} \quad (2.1)$$

where  $x_0 = 10^{-7} \text{ m}/\sqrt{\text{Hz}}$  for the GEO 600 site. Hence, with a noise specification of  $7 \times 10^{-20} \text{ m}/\sqrt{\text{Hz}}$  at 50 Hz, an isolation from ground motion  $\sim 6 \times 10^9$  is required.

### 2.3 The Mechanics of Passive Vibration Isolation

Typically, vibration isolation can be thought of as something soft (“springy”) interfaced between a source of vibration and an object requiring isolation. The Mass–Spring system exemplifies this.

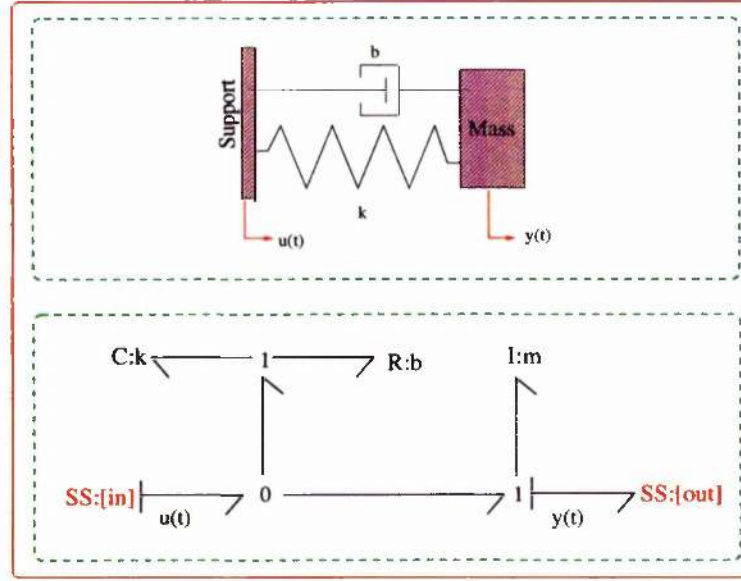


Figure 2.2: Schematic and bond graph of a Mass–Spring isolation system

$$G(s) = \frac{bs + k}{ms^2 + bs + k} \quad (2.2)$$

A schematic, and bond graph<sup>1</sup>, of a Mass–Spring system can be seen in Figure 2.2. The transfer function  $G(s)$ , Equation 2.2, of this Mass–Spring system can be generated in the usual manner, i.e. via the Laplace transform of the second order differential equation or via a suitable transformation of the core bond graph model (see Appendix A). From the bond graph representation an undamped transfer function can be obtained by either removing the resistive component **R** or by setting the value of its coefficient ( $b$ ) to zero. (The generation of Bond Graphs, and their subsequent manipulation, is covered in Chapter 3).

In the undamped case Equation 2.2 is often written as

$$G(s) = \frac{\omega_n^2}{s^2 + \omega_n^2} \quad (2.3)$$

where the complex frequency  $s = \alpha + j\omega$ , and the square of the resonant (or natural) frequency  $\omega_n^2 = \frac{k}{m}$ ,  $k$  is the spring constant and  $m$  the mass.

<sup>1</sup>Since the bond graph methodology is extensively used throughout this thesis the bond graph model of a Mass–Spring system has been included here for completeness.

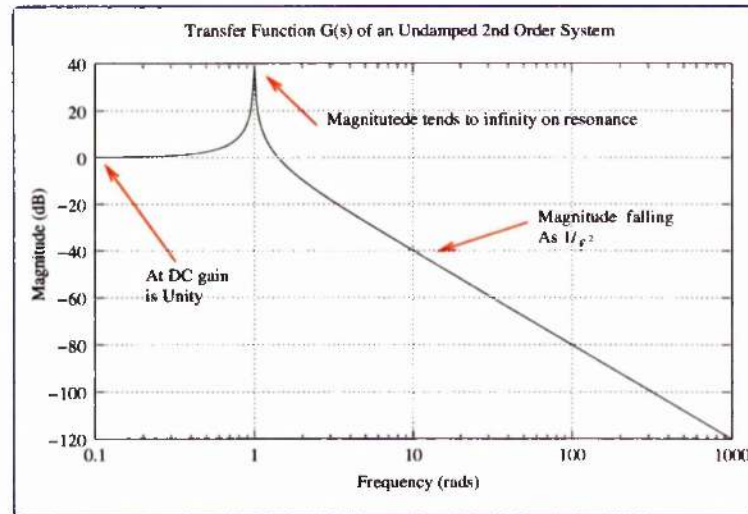


Figure 2.3: *The transfer function of an undamped second order system*

The characteristics of an undamped second order transfer functions can be seen in Figure 2.3

Points to be drawn from this transfer function are:

- From dc, to the resonant frequency, the system's frequency response approximates to unity.
- At the resonant frequency the magnitude tends to infinity.
- At frequencies above the resonance the system response falls as  $1/f^2$ , and hence, above its resonant frequency, a second order undamped system affords isolation of  $1/f^2$ .

This analysis assumes no damping, i.e. energy, once supplied to the system, remains without loss. For real systems this is not the case and energy will dissipate by various mechanisms. These include friction, as well as viscous and structural damping. A discussion of a damped second order system will follow an introduction to viscous and structural damping.

### 2.3.1 Viscous and Structural Damping Forces

Those resistive forces which are proportional to velocity are known as viscous forces. For example, a mass moving with increasing velocity through air experiences, due to increased mass/air particle collisions, an increasing resistive force which is proportional to the mass velocity. Viscous damping can be introduced into a model by the inclusion of a dashpot parallel to the components experiencing the resistive force, as shown in Figure 2.2

Spring damping can be introduced by inclusion of a small imaginary component in the spring constant  $k \rightarrow k[1 + i\phi(\omega)]$ , where  $\phi$  is positive and known as the *loss angle*, and represents a phase delay between application of force at a point and the resulting displacement of that point. If  $\phi(\omega)$  is proportional to  $\omega$  it takes the same form as viscous damping and can be modeled in a similar fashion (i.e a dashpot parallel with the spring).

However, not all cases exhibit this form of damping; some show losses that are constant across some frequency range (i.e.  $\phi(\omega) = \phi_0$  [28]). Such damping is known as *structural*. Loss angles, associated with structural damping, are typically very small; room temperature values range from  $10^{-3}$  for metals to  $10^{-9}$  for sapphire. It is from calculations using this form of damping that the thermal noise is calculated as  $7 \times 10^{-20} \text{m}/\sqrt{\text{Hz}}$  at 50 Hz [29].

### 2.3.2 The Characteristics of a Damped Second Order system

Returning to the transfer function of the damped Mass–Spring system and the viscous forms of damping, Equation 2.2, these systems are often parameterised using the damping factor ( $\zeta$ ) (Equation 2.4):

$$G(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.4)$$

where

$$\zeta = b/2\sqrt{mk} \quad (2.5)$$

and for systems with minimal damping, the system is parameterised by the quality factor  $Q = 1/2\zeta$ . The quality factor is related to the loss angle ( $\phi$ ) where  $Q = 1/\phi$ .



The system now responds in the following manner.

- Again the frequency response, up to the resonant frequency, is approximately unity.
- The resonant peak is approximately  $Q$  times the low frequency response.
- At frequencies higher than the natural frequency ( $\omega_n$ ), but less than approximately  $Q\omega_n$ , the system responds as  $(\omega_n/\omega)^2$ .
- Above  $Q\omega_n$  the magnitude of the response falls as  $1/\omega$ .

These phenomena are illustrated in Figure 2.4 which shows the transfer functions of two second order systems, one with infinite  $Q$  (no damping) and the other with a  $Q$  of 10. Here it should be noted that very high  $Q$  second order systems maximise the above resonance fall off.

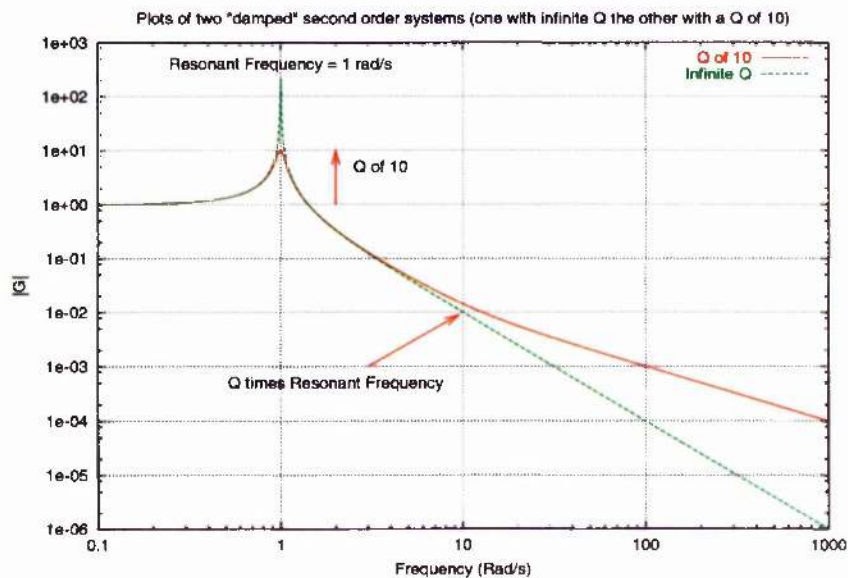


Figure 2.4: The transfer functions of two 2nd order systems, one with infinite  $Q$  the other with a  $Q$  of 10

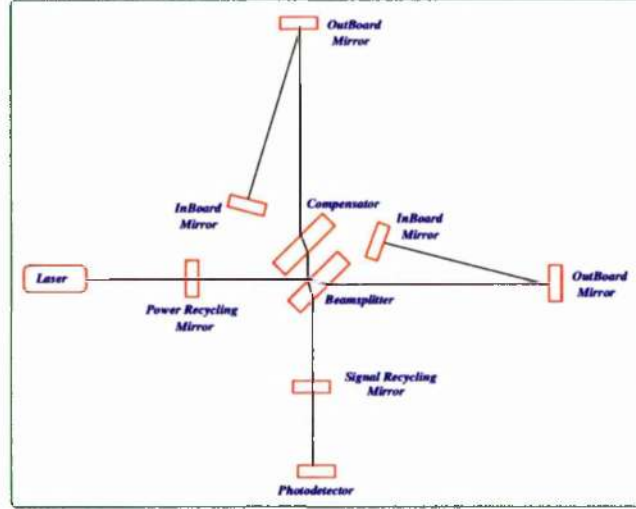


Figure 2.5: Schematic of the GEO 600 gravitational wave detector

## 2.4 GEO 600 and Passive Isolation

A schematic of GEO 600 can be seen in Figure 2.5 which is based upon a Michelson Interferometer (as described in Section 1.3.3). Each arm of the interferometer is 600 metres long and each optic is suspended as a three stage pendulum. The detector is enclosed within a vacuum system at a pressure of  $\sim 10^{-8}$  mbar. The vacuum system affords acoustic noise isolation and suppresses refractive index fluctuations. Suspending the optics as pendula affords isolation from seismic noise. Additional vertical and horizontal isolation is provided by passive and active isolation stacks and cantilever springs provide extra vertical isolation. The schematic of a main GEO 600 suspension ( Figure 2.6) illustrates the location of each of these elements.

### 2.4.1 Passive Isolation stacks

Passive isolation stacks consist of three rubber (RTV 615) bungs supporting a circular aluminium block, all of which is encased by damped bellows to prevent contamination of the vacuum. The isolation stacks are designed to have a horizontal resonant frequency at  $\sim 9$  Hz and a vertical resonant frequency at  $\sim 13$  Hz with Qs of approximately 10 [30] [31].

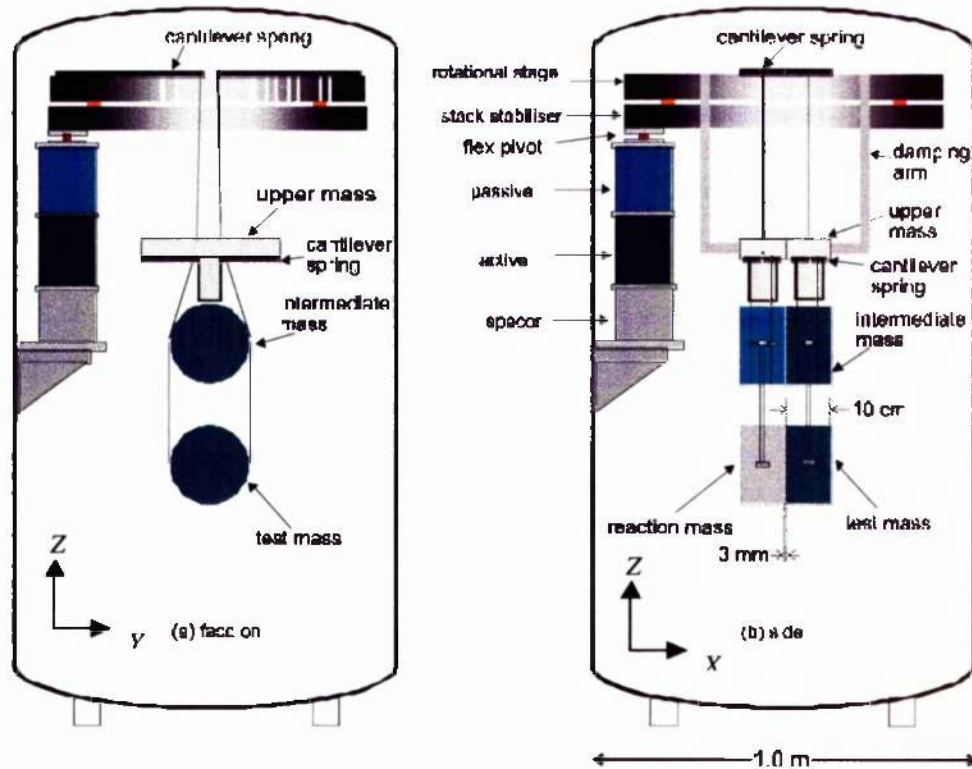


Figure 2.6: Schematic of the GEO 600 main suspension [2]

The passive isolation stacks, detailed above, are augmented with an active stage. This active stage consists of a piezo-electric transducer (PZT) and a number of geophone sensors. Here feedback and feedforward control laws will be employed to further reduce the transmittance of seismic noise from the ground to the isolated mass.

#### 2.4.2 Increased Vertical Isolation using Cantilever Springs

Remember GEO 600 is a 4 pass delay line interferometer. This means that laser light traverses the length of one arm four times and thus increases the effective arm length by a factor of two. To achieve this, light is reflected from the end mirror to an inboard mirror, at which point the light is reflected back to the end mirror and then onto the beamsplitter. Figure 2.5 illustrates this.

To facilitate the inclusion of a delay line the inboard mirror is raised above the end mirror

by  $2h = 30$  cm and hence needs to be angled by an amount equal to  $h/l$ . This results in cross-coupling between vertical and horizontal motion [32]. Moreover, due to the curvature of the earth, the “local vertical” at each mirror location is not parallel. As a consequence pure vertical motion, at a particular suspension, will couple into horizontal motion as “seen” by the laser beam. The amount of cross coupling is given by

$$\frac{X_{\text{beam}}}{Y_{\text{mirror}}} = \frac{L/2}{r_{\text{Earth}}} \quad (2.6)$$

For GEO 600 this corresponds to a cross coupling of  $5 \times 10^{-5}$ . Further, due to material defects and flaws in mechanical construction, cross couplings, between all degrees of freedom, will occur. Work by Husman [33] has shown that, with due diligence, a cross coupling  $< 0.01\%$  due to misalignments can be expected. Hence the assumption, adopted by the GEO 600 project, that a total cross coupling of less than  $0.1\%$ , from vertical to horizontal motion, is thought to be non-conservative.

With this assumption in mind, what effect does this have on the horizontal isolation at 50 Hz? Assuming a horizontal resonance at 1 Hz and a vertical resonance at 20 Hz (not unrealistic values for pendulums used within GEO 600), and assuming the vertical and horizontal seismic noise inputs are of the same magnitude, then for a triple pendulum

$$\left(\frac{0.1}{100}\right) \left(\frac{20}{50}\right)^6 > \left(\frac{1}{50}\right)^6 \quad (2.7)$$

and hence the motion due to the coupled vertical motion is greater than that from horizontal motion, i.e. greater vertical seismic isolation is needed if the required sensitivity is to be met. This is achieved through the incorporation of cantilever springs. The blades used by GEO 600 are based on those used by the VIRGO group [34]. Those used within GEO 600 are detailed in [35] [30] [33]. These cantilever springs are constructed from pre-stressed maraging steel of a trapezoidal shape and act as soft springs. Prior to loading, their shape is as shown in Figure 2.7 and once loaded are horizontal. For this trapezoidal geometry the spring constant is given by

$$k_{\text{cant}} = \frac{Eah^3}{4l^3\alpha} \quad (2.8)$$

where,  $E$  is the Young’s modulus for the cantilever material,  $a$  is the cantilever’s base width,



$h$  the thickness,  $l$  the length and  $\alpha$  is a geometric factor given by

$$\alpha = \frac{3}{2(1-\beta)} \left( 3 - \frac{2}{1-\beta} \left( 1 + \frac{\beta^2 \ln(\beta)}{1-\beta} \right) \right) \quad (2.9)$$

where  $\beta = b/a$ , and  $b$  is the width at the tip of the blade. The value of  $\alpha$  range from 1.0 for a rectangle to 1.5 for a triangle. The value of  $\alpha$  associated with GEO 600 blades was calculated and verified experimentally by Husman and Torrie [33] [30].

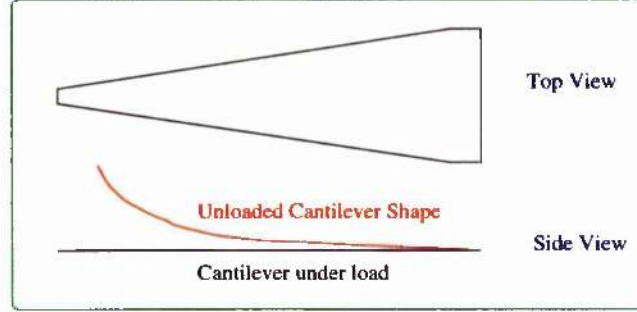


Figure 2.7: Schematic of a cantilever spring

### 2.4.3 Pendulum Suspension

A simple pendulum has a similar transfer function to that of the Mass–Spring system of Section 2.3, (Equation 2.3 and Equation 2.4 - undamped and damped responses respectively), and hence has similar vibration isolation properties. For a pendulum the natural frequency ( $\omega_n$ ) is given by

$$\omega_n = \sqrt{\frac{g}{l}} \quad (2.10)$$

where  $g$  is the acceleration due to gravity and  $l$  is the pendulum's length. Resonant frequencies on the order of 1 Hz are easily achieved. Moreover, pendulum restoring forces are dominated by the lossless gravitational force. Hence potential energy is predominately stored in the gravitational field with a small fraction stored in the bending and stretching of the wire, this relationship is characterised by the “dilution factor” [36]. Hence, pendula are ideal as seismic isolation stages. Moreover, constructing the pendula from very high Q materials ( $\sim$  a few  $10^6$ ) reduces the off resonance displacement noise due to thermal noise.

This is because, as the  $Q$  of a system increases, thermal energy is increasingly confined to narrower frequency bands centred on the resonant frequencies. Moreover, high  $Q$  cascaded passive elements provide isolation of

$$\frac{1}{\omega^{2n}} \quad (2.11)$$

above the highest resonant frequency, where  $n$  is the number of passive elements. For detector sensitivity requirements, and the fact that very low resonant frequency pendulums are very long, a multi-stage pendulum design is used. Moreover, since passive isolation elements do not require energy sources, and do not provide energy to the system, they are intrinsically stable.

There are a number of disadvantages with using passive elements. The first is that their physical size tends to increase as the resonant frequency is reduced, which may make their inclusion impracticable. Further, as stated earlier, pendulums will resonantly enhance seismic noise and, because high  $Q$  materials are used in the construction of these pendula, this resonantly enhanced motion will take many cycles to decay. To prevent this phenomenon from limiting the sensitivity of the detector these pendula are actively controlled. The aim of this control scheme is to damp the motion of the pendula such that their effective  $Q$  is reduced to a minimum. This method of reducing the effective  $Q$  does not increase the off resonant thermal noise because the energy is removed from the system externally rather than internally (frictional losses are internal losses and hence contribute to the overall noise signature). Since this control scheme acts on individual multi-stage pendula it is known as *local control*. The specifics of this form of control are described in Chapter 6. However prior to introducing this control scheme the modelling of multi-stage pendula using the bond graph methodology, is presented. These models will subsequently form the basis for a bond graph controller design.

## **Chapter 3**

# **Bond Graphs**

### **3.1 Introduction**

Following an exposition of the modelling philosophy this chapter introduces basic bond graph components, constitutive relationships and the concept of causality. It is not an exhaustive review, and is only intended as an introduction to the bond graph methodology. Comprehensive works on bond graphs are readily available [3] [37] [38]. Also a comprehensive resource list of bond graph material is available at the Glasgow University bond graph mirror site [39].

### **3.2 Modelling Philosophy**

A physical system may be represented by a multitude of different models. The type and complexity of any model will depend upon its end use. Naturally, no model can exactly replicate a physical system. Hence, the key factor in the development of a model, is the level of complexity needed to answer the modeller's questions – Too detailed and it may be impossible to analyse and extract essential information from the array of parameters – Too simplified and the model will not reveal essential system information. The aim of the modeller should be to produce the simplest model capable of supplying the relevant infor-

mation. For example, in the gravitational wave detector the violin modes of the suspension fibres are irrelevant in the design of the local control feedback. To include them would produce unnecessarily high order models and hence high order controllers, making them more sensitive to parameter errors. Yet, not to include them whilst developing the global control would result in an unstable controller. Ideally then, a *core* model of a system from which various representations can easily be extracted would facilitate the development of system models of various complexities and for different end uses. Obviously, if the core model is hierarchical it enables subsystem components to be changed without having to re-model the complete system. Furthermore, if the modelling technique is unambiguous it can be understood by a computer program and hence the power of modern computers can be utilised to extract, via model transformations, specific model representations. Bond graphs meet these criteria and are therefore ideal for modelling the multi-stage pendula used within the gravitational wave detector.

### 3.3 An Overview of Bond Graphs

Bond graphs are graphical representations of physical systems constructed in the energy domain. As such, they provide a concise method for the conveyance of much system information. By using a small set of idealised elements, system models in domains such as electrical, mechanical, and hydraulics can easily be produced. Moreover, since these same elements are used across all domains, multiple domain systems can be readily constructed. Furthermore, it is possible to use the bond graph as a core representation of the system from which other representations, such as state space equations and transfer functions, can be generated. Also, since the bond graph representation is unambiguous, a computer can be utilised to carry out these transformations. In addition, the bond graph methodology is equation based rather than assignment based. This reduces the need for multiple models of a single physical system. For example, the assignment based methodology requires two simple resistor models; one for each input. That is,  $v := Ri$  if the input is current and  $i := R^{-1}v$  if the input is voltage. As can be appreciated, the number of component models increases rapidly with the number of component input/output permutations. However,

in general, only one bond graph model is required for each component. Hence, a generic system model can usually be produced without a priori knowledge of a subcomponent's input/output configuration. Moreover, since the methodology is modular, complex system models can be constructed in a hierarchical manner. Thus the reticulation of a system into its component parts allows the system model to be constructed from simple subsystems where the physical laws are understood.

### 3.4 Elementary Bond Graph Components

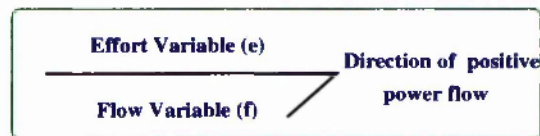


Figure 3.1: A simple bond

Hierarchical bond graphs are constructed from a few elementary components. The simple energy bond is illustrated in Figure 3.1. The half arrowhead points in the direction of positive energy flow. Associated with each bond are two *co-energy* variables the product of which gives power. These are generally known as the effort ( $e$ ) and flow ( $f$ ) variable (an alternative is across and through – in this thesis the former is exclusively used). It is common practice for the half arrow to be placed on the side of the bond associated with the flow variable. Table 3.1 shows effort and flow variables for various energy domains. Throughout this thesis the mechanical domains (rotational and linear) are exclusively considered. However, all the underlying theory is equally applicable across other domains. A set of simple components form the basis of the bond graph methodology. Each component has an input/output relationship defined by its *constitutive relationship* (**CR**). These constitutive relationships relate input variables to output variables and as such may be linear or nonlinear. Points at which energy flows from one bond graph element, or subcomponent, to another are known as *ports*. The simple one port, two port and multiport structural elements provide the basic building blocks for producing compound subcomponents. These will now be detailed, more complex components and associated constitutive relationship

Domain	Effort	Flow
Mechanical Translational	Force	Velocity
Mechanical Rotational	Torque	Angular Velocity
Electrical	Voltage	Current
Hydraulic	Pressure	Flow Rate

Table 3.1: *Effort and flow variables by domain*

will be introduced as necessary.

### 3.4.1 One Port Components

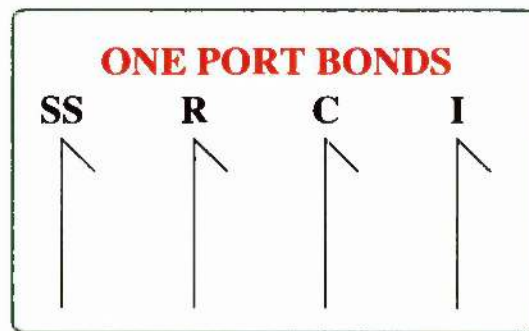


Figure 3.2: *The set of one port bond graph components: The source sensor **SS**, energy dissipator **R** and the two energy storage ports **C** and **I***

There are four elementary one port components (Figure 3.2): One source/sensor (**SS**), one energy dissipater (**R**) and two energy storage ports, (**C** and **I**),

#### The Source/Sensor Component (**SS**)

The **SS** component is used to interface bond graph subcomponents and define boundary conditions. Moreover, **SS** components are used as source elements to supply power to a



system. A source component holds one of the power variables (effort or flow) constant or is some pre-defined function of time. The value of the co-variable is determined by the system the source supplies. Remember, the half arrow of a bond points in the direction of positive power flow, and hence when  $f(t)e(t)$  is positive, energy flows from the source into the system. It should be noted that these sources are ideal in so far as the co-variable is unbounded and hence a bond graph source component may provide infinite power (real batteries are not ideal voltage sources but are generally modelled as such). Often, in bond graph literature, effort source components are represented by the symbol  $S_e$  and the flow source by the symbol  $S_f$ , see Figure 3.3. In the mechanical domain these equate to force and velocity sources respectively.




Bond Graph Symbol	Name	Defining Relationship
SS 	Arbitrary Source Sensor Symbol	
$S_e$ 	Effort Source	$e(t)$ Defined $f(t)$ Arbitrary
$S_f$ 	Flow Source	$f(t)$ Defined $e(t)$ Arbitrary

Figure 3.3: *The SS Component: effort and flow sources*

### The Resistive Component (R)

The R component, dissipates energy from a system, and has the following constitutive relationship

$$e = \Phi_R(f) \quad (3.1)$$

in the linear case  $\Phi_R(f) = Rf$ , and in the non-linear case  $\Phi_R$  is a more complex function of the effort ( $e$ ) and flow ( $f$ ) variables.

## One port energy stores

The **C** and **I** components are known as the flow and effort stores respectively. In the mechanical domain the **C** component is associated with potential energies and the **I** component with kinetic energies. Each component has an associated *energy* variable: *generalised displacement* (**q**), for the **C** component, and *generalised momentum* (**p**), for the **I** component.

### The Capacitive Component (C) or Flow Store

The flow store has the following constitutive relationship

$$e = \Phi_C(q) \quad (3.2)$$

where

$$q = \int f \, dt + q_0 \quad (3.3)$$

in the linear case

$$\Phi_C(q) = \frac{q}{C} \quad (3.4)$$

where, **C** is known as the *capacitance*. For example, a spring which obeys Hooke's law can be modelled as a linear **C** component, see Table 3.2.

### The Inertive Component I or Effort Store

The effort store has the following constitutive relationship

$$f = \Phi_I(p) \quad (3.5)$$

where

$$p = \int e \, dt \quad (3.6)$$

In the linear case

$$\Phi_I(p) = \frac{p}{I} \quad (3.7)$$

where, **I** is known as the *inertance*. An **I** component can be used to model the translational motion of a mass element, see Table 3.3.



	Generic	Spring	Variable Names
Constitutive Relationship	$e = \Phi_C(q)$	$F = \Phi_C(x)$	$e = \text{effort}$ $f = \text{flow}$
Where	$q = \int f dt$	$x = \int v dt$	$v = \text{velocity}$
Linear Case	$e = \frac{q}{C}$	$F = kx$	$q = \text{generalised displacement}$
Power	$ef$	$Fv = kx \frac{dx}{dt}$	$x = \text{displacement}$
Energy Stored	$\int e f dt$	$\frac{1}{2} k x^2$	$F = \text{force}$ $k = \frac{1}{C} = \text{spring const}$

Table 3.2: *Bond Graph capacitive component (C) modelling a spring. i.e. a store of potential energy.*

### 3.5 Structural Elements

The energy conserving structural elements provide interconnections between bond graph components. There are two *Junction* elements, the **1** or “*Series*” junction and the **0** or “*Parallel*” junction, and two *Couplers*, the *Transformer* (**TF**) and the *Gyrator* (**GY**).

The structural elements have a common constitutive relationship defined by:

$$e_1 \cdot f_1 + e_2 \cdot f_2 + \cdots + e_n \cdot f_n = 0 \quad (3.8)$$

where the subscripts 1, 2  $\cdots$  n indicate the number of ports at a junction. Note that bonds with half arrows directed towards a junction are deemed to be positive and those directed away from the junction negative.

	Generic	Mass	Variable Names
Constitutive Relationship	$f = \Phi_I(p)$	$v = \Phi_I(P)$	$e = \text{effort}$ $f = \text{flow}$
Where	$p = \int e dt$	$P = \int F dt$	$v = \text{velocity}$
Linear Case	$f = \frac{p}{I}$	$v = \frac{P}{m}$	$p = \text{generalised momentum}$
Power	$ef$	$Fv = v \frac{dP}{dt}$	$P = \text{linear momentum}$
Energy Stored	$\int e f dt$	$\frac{1}{2}mv^2$	$F = \text{force}$ $m = \text{mass}$

Table 3.3: A Bond Graph inductive component (**I**) modelling a simple mass element. i.e. a store of kinetic energy.

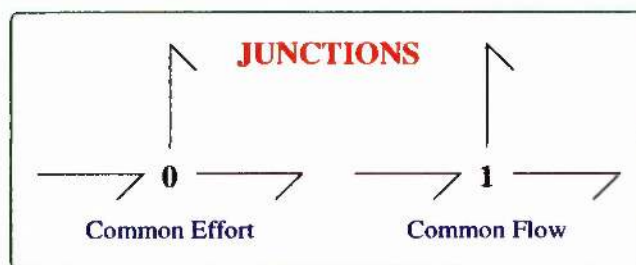


Figure 3.4: The **0** and **1** bond graph junctions

### 3.5.1 Junctions

The two junction elements (**0** and **1**) are used to interconnect bonds. Examples of three port **0** and **1** junctions are shown in Figure 3.4. The **0** junction, also known as the common effort

or parallel junction, has the following defining relationship:

$$e_1 = e_2 = \dots = e_n \quad (3.9)$$

where,  $n$  is the number of ports connected to the junction and,

$$\sum_{i=1}^n f_i = 0 \quad (3.10)$$

where,  $i$  represents the  $i^{\text{th}}$  bond attached to the junction, and the sign of  $f_i$  is positive if the half arrow of the  $i^{\text{th}}$  bond points towards the junction, and is negative otherwise.

The **1** junction, also known as the common flow or series junction compliments, the **0** junction and has the following defining relationships:

$$f_1 = f_2 = \dots = f_n \quad (3.11)$$

where,  $n$  is the number of ports connected to the junction and,

$$\sum_{i=1}^n e_i = 0 \quad (3.12)$$

where,  $i$  represents the  $i^{\text{th}}$  bond attached to the junction, and the sign of  $e_i$  is positive if the half arrow of the  $i^{\text{th}}$  bond points towards the junction, and is negative otherwise.

It should be noted that, since all the bonds at a parallel junction have a common effort, only one bond can impose an effort input. Conversely, since all the bonds at a series junction have a common flow, only one bond can impose a flow input. This idea will be made clearer in Section 3.6 (Causality) and is illustrated in Figure 3.6 (Page 36).

### 3.5.2 Coupling Elements

The bond graph representations and defining relationships for the two coupling elements, transformer and gyrator, can be seen in Figure 3.5. These are energy transforming components; in each case the effort or flow variable is transformed and hence, to conserve the energy relationship of Equation 3.8 the co-variable must also be transformed, i.e. at every instance in time

$$e_1(t)f_1(t) = e_2(t)f_2(t) \quad (3.13)$$

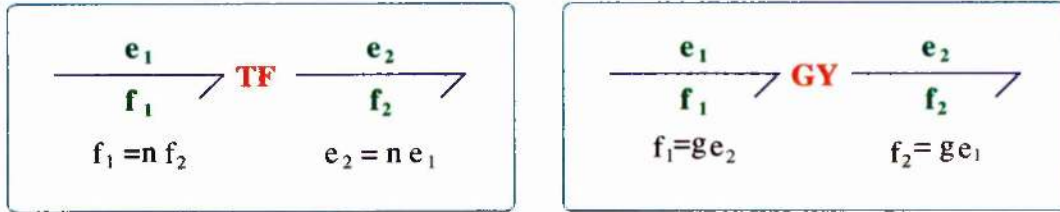


Figure 3.5: *Bond graphs for the transformer and gyrator*

The transformer **TF** has the defining relationship

$$f_1 = n f_2, \quad e_2 = n e_1 \quad (3.14)$$

and is an idealised transformer which is used to model such devices as the electric transformer and the ideal rigid lever.

The gyrator **GY** with defining relationship

$$f_1 = g e_2, \quad f_2 = g e_1 \quad (3.15)$$

relates the input flow to the output effort. The gyrator is used to model such devices as transducers and the toy gyroscope.

The coefficients  $n$  and  $g$  may have modulating coefficients, i.e the coefficient is a function of some other system variable. Modulated transformers occur regularly within the mechanical domain where the modulating terms are  $\cos(\theta)$  and  $\sin(\theta)$ , where  $\theta$  is the angular rotation. These *modulated transformers* have the symbol **MTF** (and similarly the *modulated gyrator* has the symbol **MGY**).

Combinations of the coupling elements, **TF** and **GY**, and junction elements, **0** and **1**, are known as *junction structures*. The properties of these junction structures will be detailed after an introduction to causality.

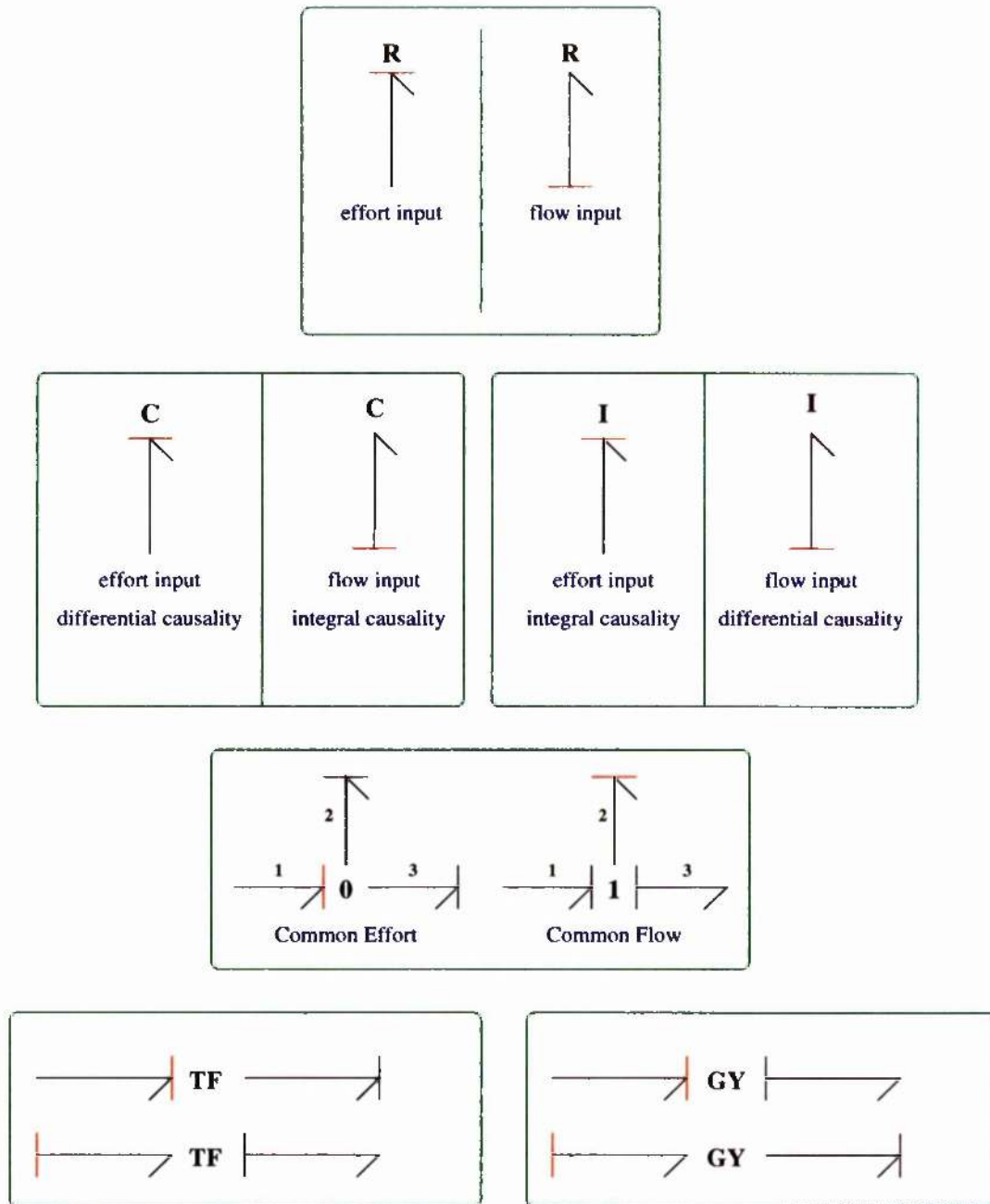


Figure 3.6: Causality for the one port components

### 3.6 Causality

Causality is a key concept within the bond graph methodology. Causality determines which variable, effort or flow, is the input and hence which is the output of a bond. A bond's



causality is defined by its causal stroke as indicated in Figure 3.6. The end of the bond with the causal stroke denotes the direction in which the effort variable is directed. For example (referring to Figure 3.6) bond 1 of the 0 junction has a causal stroke indicating that the effort variable is directed towards the junction. The assignment of a causal stroke to a bond implies the variable (effort or flow) for that bond is known and hence can be propagated throughout the bond graph. In Figure 3.6 causal strokes displayed in red are assumed to be imposed and thus define, via component constitutive relationships, all other causal strokes associated with that component (black strokes).

Causality pertaining to energy stores has a fundamental role in determining overall system equations. Assignment of the causal stroke to an energy port will result in either integral or differential causality. Integral causality arises when an energy store's input variable is integrated. Conversely with differential causality the input variable is differentiated. The causal strokes and resultant input/output relationships for the energy stores are shown in Figure 3.6 and Table 3.4 respectively. It is clear that integral causality arises when the state of the energy store is the time integral of the input variable. Perhaps the naming of the stores

Component	Integral Causality	Differential Causality
<b>C</b>	$e = \Phi_C^{-1}(\int f dt)$	$f = \frac{d}{dt}\Phi_C(e)$
<b>I</b>	$f = \Phi_I^{-1}(\int e dt)$	$e = \frac{d}{dt}\Phi_I(f)$

Table 3.4: *Integral and differential causal relationships for the C and I components*

is now clearer, in that the effort store (**I** component) has integral causality when the effort is the input, and the flow store (**C** component) has integral causality when the flow variable is the input. Integral causality on an energy bond implies that this energy store is independent of all others and it will provide one *state* to the overall system model. Conversely an energy store with differential causality is dependent on at least one other energy store and provides one *non-state* to the system model: A non-state is a function of at least one other system state.

Whilst attempting to complete the causality of a bond graph model, each source should be

taken in turn, and its associated causality propagated as far as is possible through the model. Following this, any energy stores that have not been assigned causality should be taken, one at a time, assigned integral causality, and then appropriate causal strokes propagated. Should a causal conflict arise when propagating integral causality then that component's causality must be changed to differential, and the new causality propagated. After the energy components have had causality assigned any remaining **R** components without causal strokes should have causality arbitrarily assigned. Generally these are assigned to reduce mathematical difficulties (singularities etc). On completion of the above process one of three outcomes will occur:

1. **The bond graph is under causal** (incomplete causality). This implies that system variables can not be explicitly computed from system inputs and component constitutive relationships. That is, algebraic loops exist and therefore simultaneous equations must be solved. (System algebraic equations are not lower triangular).
2. **The bond graph is over causal** (conflicting causality). Here two components attempt to cause the same variable and the model is not physical. If a model of a real physical system is to be created, defining new variables and additional constraints may lead to a system model.
3. **Causally complete** (or causal). The systems algebraic equations are lower triangular and hence no causal conflicts exist.

Since the causally complete bond graph is unambiguous the extraction of alternative representations (Section 3.8) can be achieved through the use of a suitable software program. Hence the modeller can concentrate on the modelling process and leave the "handle turning" to the computer. Throughout the course of this research the bond graph modelling tool MTT was utilised. This is a very powerful tool for the construction and analysis of bond graphs [40].

It should be noted that alternative software programs exist for the construction and manipulation of bond graphs. These include Bondlab, CAMBAS, Camp-G/ALSL, Dymola, a Java Applet program, Hyber Sim, MS1, 20-sim. More information concerning these software

packages can be found via the Glasgow University bond graph mirror site [41] and a bond graph dedicated internet site [42]. Throughout this thesis MTT was solely used for the construction and manipulation of bond graphs.

### 3.6.1 Junction Structures

Junction structures are the assemblages of the structure elements, i.e. the common effort (0) and common flow (1) junctions, and the transformer (TF) and gyrator (GY) coupling elements (including modulated coupling elements). A simple junction structure can be seen in Figure 3.7

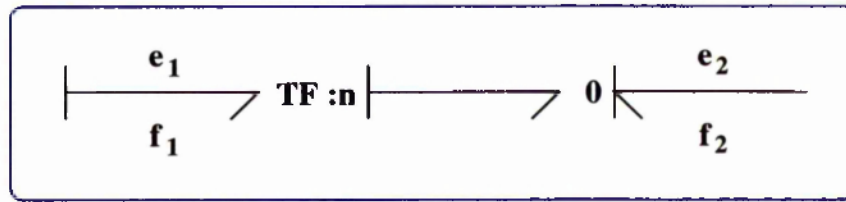


Figure 3.7: An example of a simple junction structure [3]

With the causality shown in Figure 3.7, the expressions relating the flow and effort variables of the inputs and outputs can be expressed as a matrix (Equation 3.16)

$$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & n \\ -n & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix} \quad (3.16)$$

Now, since this is an energy conserving construct the matrix relating inputs to outputs must be antisymmetric; that is, zeros must appear on the main diagonal, and the  $ij$ th component must be the negative of the  $ji$ th component. Here it should be appreciated that the junction structure relates input and output variables via a matrix transformation. Notice, that the structure elements have been used to create an appropriate transformation between input and output variables analogous to the two port **TF** and **GY** components of Section 3.5.2. These matrix transformations play a prominent role in the formation of *multiport transformers*[43]. This is best illustrated with an example:

In Figure 3.8 a multiport transformer can be seen



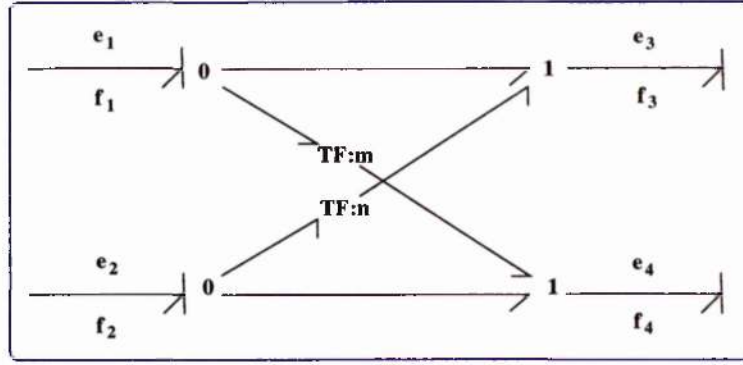


Figure 3.8: An example of a multiport transformer

Here the effort variables at the external ports are related by

$$\begin{bmatrix} e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} 1 & m \\ n & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \quad (3.17)$$

and the flows are related by

$$\begin{bmatrix} 1 & n \\ m & 1 \end{bmatrix} \begin{bmatrix} f_3 \\ f_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (3.18)$$

The important point here, is that the matrix relating the flow variables is simply the transpose of the matrix used to relate the effort variables. In this example, the transformation was performed on a two input two output multiport transformer, but generalises to any number of ports. In the general case there will be  $m$  input ports and  $n$  output ports, and the transforming matrix will be an  $n \times m$  matrix and its transpose  $m \times n$ . It should be noted that these transformers are energy conserving and hence *multiport modulated transformers* in which the elements of the matrix change with time may also be constructed [43]. This is particularly useful in mechanics, where multiport modulated transformers are used to perform nonlinear geometric transformations. For example the following relationship holds for transforming flow variables from polar to translational coordinates

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} \quad (3.19)$$

and hence, we know that the relationship between the efforts (forces and torques) simply

involves the transpose of the matrix in Equation 3.19[44], i.e.

$$\left[ \begin{array}{c|c} \cos \theta & \sin \theta \\ \hline -r \sin \theta & r \cos \theta \end{array} \right] \left[ \begin{array}{c} F_x \\ F_y \end{array} \right] = \left[ \begin{array}{c} F_r \\ \tau \end{array} \right] \quad (3.20)$$

Here it should be appreciated that if a bond graph multiport transformer can successfully be created from a valid flow (effort) transforming matrix then the efforts (flows) will, via the transpose of the original matrix, be correctly transformed. This property is implicitly used in the construction of the **CoordTrans** component (Section 4.5.3) which is used to define the location of suspension points associated with particular 2-dimensional masses.

This completes this introduction to the bond methodology and a simple example will now be presented to consolidate the ideas expounded above.

### 3.7 An Example to Illustrate Bond Graph Concepts

The aim of this section is to take a simple compound component and use it to illustrate the key concepts of the bond graph methodology. To this end the **INTF** component will be used. This component is widely used in the modelling of masses and wires where it is used to generate the angle through which components have rotated. That is the **INTF** component is a simple compound component used to integrate the flow variable.

#### 3.7.1 The need for a INTF component

As indicated in Section 3.5.2, modulated transformers can be used to provide geometric transformations. These modulated transformers require the angle through which a mass element has rotated. Within the domain of rotational mechanics the co-energy variables are torque (effort) and angular velocity (flow). Hence, time integration of the angular velocity will yield the angle. Since a bond has both an input and output it is necessary to ensure that the generation of the angular variable does not effect the system model in an undesirable manner. This is achieved through the incorporation of the amplified flow component (**AF**)

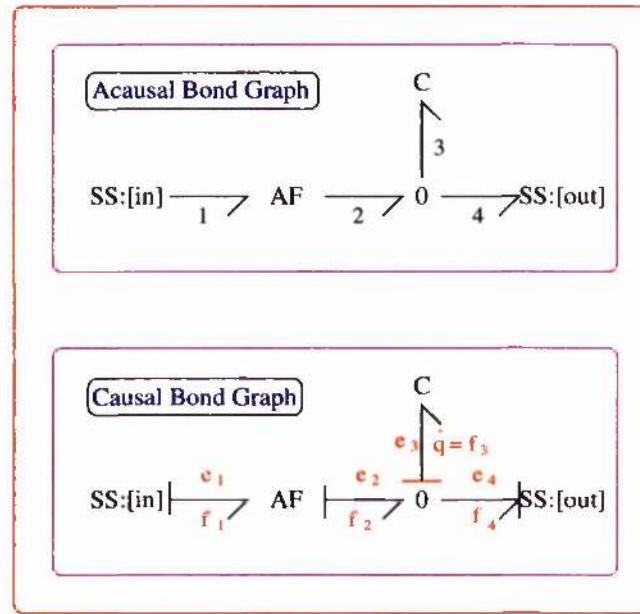


Figure 3.9: The **INTF** component: Integrates the flow variable

that prevents the transmission of the effort variable to the system model. These characteristics are embedded within the **INTF** component which will now be analysed.

### 3.7.2 INTF bond graph design

The *acausal* and causal bond graphs of the **INTF** component can be seen in Figure 3.9. An acausal bond graph is a bond graph representation prior to the assignment of causal strokes, whereas the causal bond graph is a causally complete bond graph (see Section 3.6). Note, with the causality shown in Figure 3.9 it is assumed that SS:[in] is connected to a **1** junction which contains a port whose imposed flow variable is to be integrated. That is it imposes the angular velocity for integration. The analysis of this **INTF** component is presented in Table 3.5.

	<p>: A <b>C</b> component with flow input (integral causality) (<math>q = \int f dt</math>) is used to generate the angle of rotation. Now, the constitutive relationship for the <b>C</b> component is <math>e = \Phi_c q</math>. Hence by setting <math>\Phi_c = 1</math> the effort variable (<math>e</math>) will be assigned the value of the energy variable (<math>q</math>) (i.e. the angle)</p>
	<p>: By attaching the <b>C</b> component and a one port source/sensor (<b>SS</b>) to a <b>0</b> junction (common effort) the angle variable can be directed to other areas of a system bond graph. Notice, the <b>C</b> component with integral causality imposes its effort variable (<math>e_c</math>) upon the <b>0</b> junction. Hence any further bonds attached to this junction will have <math>e_c</math> imposed upon them. Further, all other bonds must provide flow inputs. Hence the <b>SS</b>:[out] must be assigned the causality shown, as must all further bonds attached to this junction. Now, the algebraic sum of all flow variables at a <b>0</b> junction sum to zero. This leads to <math>-f_c - f_{ss[out]} = 0</math>. Notice, the signs of the flow variables: Since each bond's half arrow is directed away from the junction they are assigned a negative value.</p>
	<p>: The one port <b>SS</b>:[in] provides the flow variable for integration. Now at the <b>0</b> junction <math>-f_c + f_{ss[in]} - f_{ss[out]} = 0</math> i.e. the flow input to the <b>C</b> component <math>f_c = f_{ss[in]} - f_{ss[out]}</math>. However a problem now arises; since a bond has an input and an output (effort and flow), the effort variable (angle) from the <b>C</b> component will have a "back effect" upon the system model. This needs to be prevented.</p>
	<p>: The Amplified Flow (<b>AF</b>) component is a two port element with the following constitutive relationship: <math>f_2 = n f_1</math> and <math>e_1 = 0 e_2</math>. Hence, by assigning <math>n</math> the value of 1 gives <math>f_2 = f_1</math>, and therefore by inserting the <b>AF</b> component between the <b>SS</b>:[in] bond and the <b>0</b> junction the angular variable is prevented from influencing the system model in an undesirable manner.</p>
	<p>: Having inserted the Amplifier the causal strokes must be assigned as shown. This causal assignment is forced by the structural <b>0</b> junction and the amplifier's constitutive relationship. The right hand bond must provide flow input to the zero junction and therefore <math>e_2</math> is input to <b>AF</b> and <math>e_1 = 0 e_2</math>. Further the left hand bond must have its causal stroke imposing flow upon <b>AF</b>, as defined by its constitutive relationship. The <b>1</b> junction joining the <b>SS</b>:[in] bond and the two port <b>AF</b> component explicitly shows how the <b>AF</b> bond is inserted into the bond graph. However, the <b>1</b> junction's (common flow junction) constitutive relationship leads to <math>f_a = f_b</math> and since <math>\sum e_i = 0</math>, <math>e_a - e_b = 0 \Rightarrow e_a = e_b</math> (again signs arise from direction of half arrowheads). Hence, since <math>f_a = f_b</math> and <math>e_a = e_b</math> the <b>1</b> junction can be removed and the compound component reduces to that shown in Figure 3.9. So, a compound component (<b>INTF</b>) has been created which can be used within any hierarchical bond graph to generate angular and linear displacements without adversely affecting the system model.</p>

Table 3.5: Analysis of the compound **INTF** component

### 3.8 Transformations and Representations

Central to bond graph modelling is its use as a core model from which, via suitable transformations, other representations can be generated. Having created a bond graph model, and assigned causality, it is possible to create alternative representations of the physical system. For example, a state space representation of the **INTF** component can be extracted as follows. First, generate a set of ordered equations as shown in Table 3.6. When constructing such a table equations relating sources and sensors (i.e known variables) should be appended first). The state equations are then extracted by selecting the derivatives ( $f_3$ )

	Equation	Comment
1	$f_1 = f_{in}$	input (angular velocity)
2	$f_4 = 0$	ideal sensor
3	$e_3 = q$	integral causality ( $e_3 = q/c$ with $c=1$ )
4	$e_2 = e_3$	
5	$e_4 = e_2$	output (angle)
6	$e_1 = 0$	<b>AF</b> constitutive relationship
7	$f_2 = f_1$	$f_2 = n f_1$ with $n=1$
8	$\dot{q} = f_3 = f_2 - f_4$	

Table 3.6: A set of ordered equations for the **INTF** component

of the state variable ( $q$ ) and working backwards through the table of equations. For the **INTF** component there is only one energy store with integral causality and therefore the state space equations will have just one state as shown in Equation 3.21.

$$\begin{aligned} \dot{x}_1 &= f_{in} \\ y &= x_1 \end{aligned} \tag{3.21}$$

where,  $x_1$  represents the state variable<sup>1</sup> (the generalised displacement), and is equal to the integral of the input velocity  $f_{in}$ , and the output  $y$  is equal to  $x_1$  (as expected).

<sup>1</sup>Note the choice of state variables is not unique and a change of variables can be performed by an appropriate transformation [45]  $q$ . However, all state variables used within this thesis will be generalised displacements

Bond graph models are often non-linear. However, many of the useful tools for system analysis and control rely upon linear models. The models of mechanical systems, through the use of non-linear geometric transformations, are inherently non-linear and hence the non-linear to linear transformation will now be discussed.

### 3.9 Bond Graphs and Linearisation

As already stated, bond graph components, and hence system models, maybe linear or non-linear in nature. Hence it is possible, in fact generally desirable, to create non-linear models of physical systems using all the advantages that the bond graph methodology offers (hierarchical, energy based, unambiguous, etc.). These models can then, via a suitable transformation, be linearized about some operating point. This is particularly useful when modelling mechanical systems, such as multi-stage pendula, which through the use of geometric transformations, are inherently non-linear. Moreover, it is good practice to create non linear models, from non linearised subcomponents, and then linearise the total system model about some operating point, rather than create system models directly from linearised subcomponents. This is because the linearisation of individual components may result in loss of important system dynamics which are only manifest in the complete non-linear system model. The linearization process consists of two stages:

- finding the steady state
- performing the linearisation

Determining the steady state algebraic equations, for a system model, is often the most difficult of these stages. Taking a generic non-linear state space representation of a system,  $(\dot{q})$  and generalised momenta  $(p)$ . State equations will, in general, be displayed, with  $x_1, x_2 \dots x_n$  replacing individual  $p$ 's and  $q$ 's, and an accompanying legend will relate appropriate state variables to individual components/states.



i.e. an explicit set of first order differential equations, of the form:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.22)$$

$$\mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \quad (3.23)$$

where,  $\mathbf{x}(t)$  is an  $\mathbf{n}_x \times 1$  state vector,  $\mathbf{u}(t)$  is an  $\mathbf{n}_u \times 1$  vector of inputs,  $\mathbf{y}(t)$  is an  $\mathbf{n}_y \times 1$  vector of outputs. The steady state solution implies that  $\dot{\mathbf{x}} = 0$ . Thus the steady state solution corresponds to  $\mathbf{x} = \mathbf{x}_0$ , and  $\mathbf{u} = \mathbf{u}_0$  such that

$$f(\mathbf{x}_0, \mathbf{u}_0) = 0 \quad (3.24)$$

The linearised approximation of the non-linear system is achieved via a first order Taylor series expansion, about the steady state:

$$\Delta \dot{\mathbf{x}}(t) = \mathbf{A}_0 \Delta \mathbf{x}(t) + \mathbf{B}_0 \Delta \mathbf{u}(t) \quad (3.25)$$

$$\mathbf{y}(t) = \mathbf{C}_0 \Delta \mathbf{x}(t) + \mathbf{D}_0 \Delta \mathbf{u}(t) \quad (3.26)$$

where  $\Delta \mathbf{x}(t)$  and  $\Delta \mathbf{u}(t)$  are the small disturbances about the operating point of  $\mathbf{x}_0$  and  $\mathbf{u}_0$ ,

$$\Delta \mathbf{x} = \mathbf{x}(t) - \mathbf{x}_0 \quad (3.27)$$

$$\Delta \mathbf{u} = \mathbf{u}(t) - \mathbf{u}_0 \quad (3.28)$$

and the matrices  $\mathbf{A}_0$ ,  $\mathbf{B}_0$ ,  $\mathbf{C}_0$  and  $\mathbf{D}_0$  are *Jacobian matrices* evaluated at  $\mathbf{x}_0$  and  $\mathbf{u}_0$ , that is

$$\mathbf{A}_0|_{\mathbf{x}_0, \mathbf{u}_0} = \left[ \begin{array}{cccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{array} \right] \bigg|_{\mathbf{x}_0, \mathbf{u}_0} \quad \mathbf{B}_0|_{\mathbf{x}_0, \mathbf{u}_0} = \left[ \begin{array}{cccc} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \cdots & \frac{\partial f_1}{\partial u_{n_u}} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \cdots & \frac{\partial f_2}{\partial u_{n_u}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \cdots & \frac{\partial f_n}{\partial u_{n_u}} \end{array} \right] \bigg|_{\mathbf{x}_0, \mathbf{u}_0} \quad (3.29)$$

$$\mathbf{C}_0|_{\mathbf{x}_0, \mathbf{u}_0} = \left[ \begin{array}{cccc} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{array} \right] \bigg|_{\mathbf{x}_0, \mathbf{u}_0} \quad \mathbf{D}_0|_{\mathbf{x}_0, \mathbf{u}_0} = \left[ \begin{array}{cccc} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \cdots & \frac{\partial g_1}{\partial u_{n_u}} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \cdots & \frac{\partial g_2}{\partial u_{n_u}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial g_n}{\partial u_1} & \frac{\partial g_n}{\partial u_2} & \cdots & \frac{\partial g_n}{\partial u_{n_u}} \end{array} \right] \bigg|_{\mathbf{x}_0, \mathbf{u}_0} \quad (3.30)$$

Where  $\mathbf{A}_0$  is an  $n_x \times n_x$ ,  $\mathbf{B}_0$  an  $n_x \times n_u$ ,  $\mathbf{C}_0$  an  $n_y \times n_x$ , and  $\mathbf{D}_0$  an  $n_y \times n_u$  matrix.

It should be appreciated that all the bond graph multi-stage pendula models contained within this thesis are non linear. When creating alternative linear representations of the core bond graph model, via appropriate transformations (see Section 3.8), linearisation will be performed, by MTT, on the non-linear model as necessary.

### 3.10 Conclusions

In this chapter the bond graph modelling technique has been introduced and a simple compound component used to illustrate the methodology. In the following chapters the methodology will be used to construct complex hierarchical models of multi-stage pendula in a modular fashion. These models will then be used to create physical controllers designed within the bond graph domain. Which will in turn be evaluated on a physical system.



## Chapter 4

# 2-Dimensional Bond Graph Pendulum Models

### 4.1 Introduction

This chapter details the bond graph model of a multi-stage pendulum. The goal was to produce an easily extensible model that would facilitate the investigation of various system configurations and aid rapid development of control laws.

From the outset the aim was to exploit the hierarchical nature of bond graphs to produce a library of components, that a subsequent user could easily configure to produce models of various complexities. With the end user in mind, the purpose of this chapter is to detail not only the specifics of individual model components, but also to form the basis of a tutorial.

The model was developed using a bond graph manipulating software package MTT<sup>1</sup>(Model Transformation Tools). Although any bond graph component presented in this thesis is not explicitly dependent upon this software package, its use is recommended. This is because a library of pendulum components, compatible with MTT, already exist [46]. Therefore an end user already has, “at his/her finger tips”, all the necessary tools needed to investigate

---

<sup>1</sup>This, freely available software package, can be found at [www.sourceforge.net/projects/mtt](http://www.sourceforge.net/projects/mtt). It can be used with any Linux or Unix based operating system.

the behavior of multi-stage pendula. It is assumed that a user will employ MTT<sup>2</sup> to carry out their analysis, and as such the use of pendulum models will be demonstrated with this in mind.

MTT is well documented, and a manual that accompanies the software is available at the same repository as the software. As such, only those aspects of MTT usage that are directly relevant to the construction and analysis of pendulum models will be detailed here. It is assumed that relevant software packages (Reduce, Octave, xfig etc) are all available to the user and the user knows how to invoke the MTT interface. MTT can be run in a command line or (menu driven) windows environment. In this thesis MTT examples will be illustrated using the command line interface. Examples of MTT usage are presented using the `type-writer` typeset. (New users should, in the first instance, execute the command `mtt` as this will present the user with information on general MTT usage and options.

Bond Graphs are constructed using the xfig computer graphics program and have file names of the form `ComponentName_abg.fig`<sup>3</sup>, (`abg` means acausal bond graph (see Chapter 3)).<sup>4</sup> Each `ComponentName_abg.fig` file has an associated `ComponentName_lbl.txt` file. This is a *label* file that contains information about components, their constitutive relationships and associated parameters. Aliases are also declared within the `lbl` file and will be discussed later. For each of the predefined pendulum components, appropriate `lbl` files exist. Hence there should be no need for an end user to edit these files. However, if a new top level bond graph is required, say a triple pendulum, or a new mass component, then a new bond graph will need to be generated and therefore an associated `lbl` file created/edited. To illustrate the various nuances of these `lbl` files a selection will be presented within the general discussion of their associated `abg` files. Having perused these, users should have no problems creating new system models.

---

<sup>2</sup>Naturally, as MTT undergoes continuous development, certain features detailed here may have changed within the current version. However the files pertaining to the components detailed within this thesis stand alone in the sense that MTT is backwardly compatible. Hence the user may use the bond graph models contained here to gain a feel for bond graphs and MTT and then consult the latest documentation for any developments. Models contained within this thesis were developed using MTT version 4.7.3

<sup>3</sup>Bond graphs shown within this thesis are postscript versions of the original `.fig` files

<sup>4</sup>acausal is a misnomer in the sense that should a preferred causality be desired then causal strokes maybe applied by the modeller at this stage. However if no causal strokes are applied, MTT (by default) will attempt to maximise integral causality and will only prompt the user if there is a causal conflict.

## 4.2 The Model

As stated in Chapter 3 the bond graph methodology is hierarchical in nature, enabling the construction of complex systems in a modular fashion. Bond graphs can fully exploit modularisation due to the acausal nature of individual components. Remember that bond graphs are equation not assignment based. Hence, in general, one bond graph component can represent an individual system component no matter the input output configuration. This is not possible with the assignment based methodologies where an individual component is only re-usable if the causality is fixed (Causality is discussed in Section 3.6).

The elements of the 2-dimensional multi-stage pendula are detailed in the following sections. First a top level (or level-0) bond graph of a double pendulum is presented. This top level bond graph is constructed from a number of subcomponents, starting from this top level, each component (level-1) will be taken in turn and its properties, along with those of any associated subcomponents (level-2 and deeper), will be detailed.

## 4.3 A Double Pendulum Bond Graph

Figure 4.1 shows a schematic, and bond graph, of a two dimensional double pendulum. The main components of this bond graph are, **Wire**, **TwoWires**, **UpperMass**, and **TestMass**, which are in turn constructed from further subcomponents. The half arrows are vector bonds, in each case they represent three bonds carrying the effort and flow variables associated with the X, Y and angle (henceforth angle will be denoted by the symbol  $\Theta$ ). Here, X and Y are the horizontal and vertical coordinates respectively and  $\Theta$  is the angle of rotation as measured from the Y axis. The bracketed terms, [in] and [out], are vector port labels. They are markers that inform MTT as to which bonds of the subcomponent the vector bond should be associated with, and are declared within the 1b1 file of the subcomponent. For example, within the label file of **Wire** the following lines of text must exist:-

```
% ALIAS      in      x_in,a_in,y_in
% ALIAS      out     x_out,a_out,y_out
```

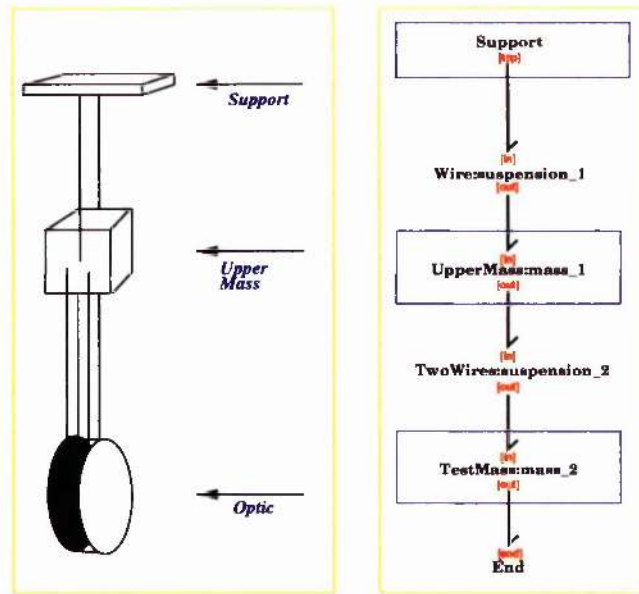


Figure 4.1: *Double pendulum schematic and bond graph*

Remember the declaration occurs in the subcomponent and denotes that an external vector bond named [in] connects to the **SS** (see Section 3.4.1) components **SS:[x\_in]**, **SS:[a\_in]** and **SS:[y\_in]** of the subcomponent **Wire**.

This is the first of three forms of aliasing and is known as *Port Aliasing* (see Figure 4.2 for a general description), the other forms are *component* and *parameter* aliasing and will be covered later.

Should a user require alternative system models a top level bond graph, similar to that shown in Figure 4.1, should be created using xfig and saved as `YourModelName_abg.fig`. Here an existing bond graph can be edited or by executing the following command:

```
prompt> mtt ComponentName abg fig
```

an xfig window will open. Within this window a new bond graph can be created from a library of MTT components – just click on the library icon, click on the “not loaded” button, and then choose the “basic” option. This will load a set of objects that facilitates the creation of consistent bond graphs. Care should be taken when constructing models, bonds

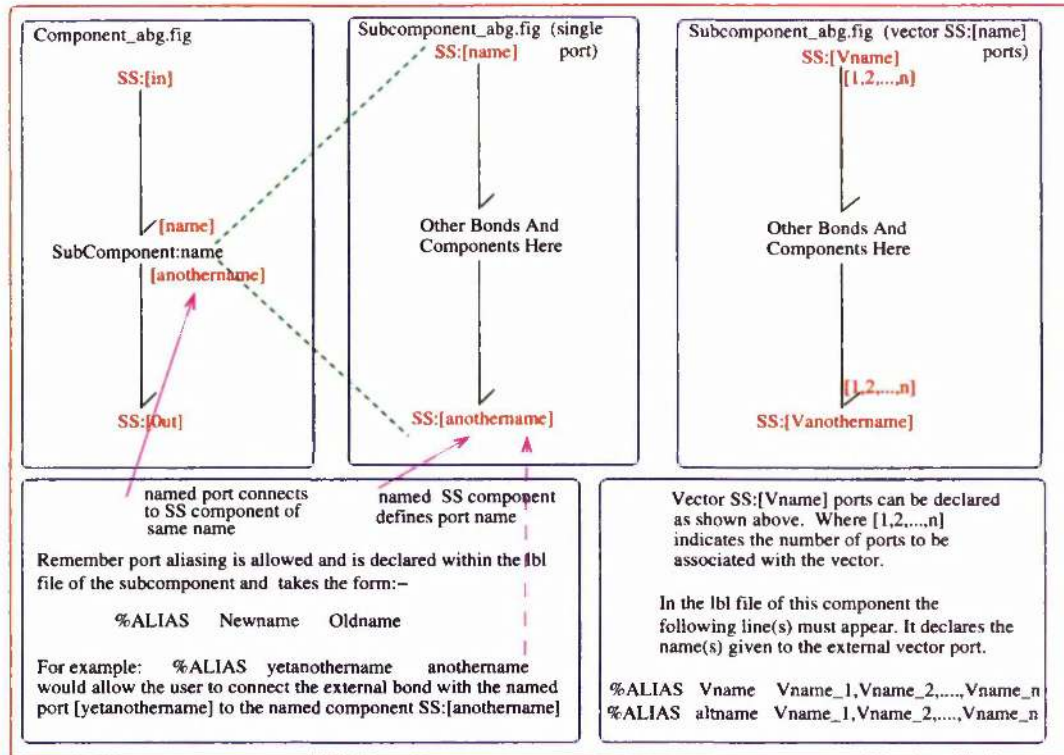


Figure 4.2: The SS bond graph component and the use of named ports, port aliasing and vector bonds

are associated with a particular component by calculating the geometric distance from a bond to a component. If the user is not careful with the placement of bonds, relative to components, ambiguities may arise.

Having generated a bond graph the user should issue the command

```
prompt> mtt ComponentName lbl txt
```

MTT will then generate an appropriate lbl file. MTT creates, not only the Component-Name\_lbl.txt file, but also a new directory MTT\_Work, this directory contains all the intermediate files generated by MTT during the execution of an MTT command. At this stage the user need not concern themselves with the details of this directory.

The label file for Double (substituting Double for ComponentName) can be seen in Ta-

```

%% Label file for system Double (Double_lbl.txt)
%SUMMARY Double
%DESCRIPTION <Detailed description here>

% =====
% %% Version control history
% =====
% %% $Id$
% %% $Log$
% =====

% Port aliases
}
% Argument aliases

%% Each line should be of one of the following forms:
%           a comment (ie starting with %)
%           component-name      cr_name arg1,arg2,..argn
%           blank
% ----- Component labels -----

% Component type UpperMass
    mass_1  none  m_1;j_1;g;hu_1;hl_1;hr_1;vu_1;vl_1;vr_1
% Component type TestMass
    mass_2  none  m_2;j_2;g;hl_2;hr_2;vl_2;vr_2
% Component type Wire
    suspension_1  none  l_1;k_1
% Component type TwoViolinWires
    suspension_2  none  l_2;k_2;

```

Table 4.1: *Double\_lbl.txt: Double pendulum label file*

ble 4.1. The text in typewriter typeset is automatically generated by MTT and the *italicised* text is supplied by the user. At this top level all the user need do is fill in the italicised text, as appropriate for individual models. As can be seen, MTT creates a new line for each component. Associated with each component are two fields; the first declares the constitutive relationship and the second is a list of parameters. Examples are given in Table 4.2.

component name	field 1 (constitutive relationship)	field 2 (parameters)
mass_1	<i>none</i>	<i>m_1;j_1;g;hu_1;hl_1;hr_1;vu_1;vl_1;vr_1</i>
suspension_1	<i>none</i>	<i>l_2;k_2</i>

Table 4.2: *Form of the information fields for a top level bond graph*

The examples given within Table 4.2 are the exact format for all component lbl files where no explicit constituent relationship is given (i.e none). The list of parameters, separated by



“;” are precisely those required to pass parameters to subcomponents. This is the simplest form of the lbl file; further nuances of the lbl file are covered later.

This fully details the top level bond graph. In the following sections individual components are detailed.

## 4.4 Support

The bond graph of an idealised pendulum support can be seen in Figure 4.3 and it’s associated label file in Table 4.3. This component is used to provide the boundary conditions at the top of the pendulum. That is, zero velocities are applied at the upper end of the pendulum suspension element. Moreover, it also provides a point at which disturbances can be injected into the model.

```
%% Label file for system Support (Support_lbl.txt)
%SUMMARY Support
%DESCRIPTION <Detailed description here>

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id$
% %% $Log$
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Port aliases
%ALIAS    in|top      x_s,a_s,y_s

% Argument aliases

%% Each line should be of one of the following forms:
%          a comment (ie starting with %)
%          component-name      cr_name arg1,arg2,..argn
%          blank

% ---- Component labels ----
% Component type SS
      x_s      SS      internal,0
      a_s      SS      internal,0
      y_s      SS      internal,0
      [x_s]    SS      external,external
      [a_s]    SS      external,external
      [y_s]    SS      external,external
```

Table 4.3: *Support\_lbl.txt: Support label file*

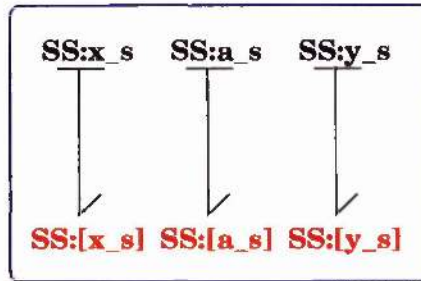


Figure 4.3: *Bond graph of an idealised pendulum support*

Points to note here are:

- 1 **SS:[i\_s]** (where  $i=x, a$  or  $y$ ) : are [named] **SS** components. These act as interfaces between the internal bonds of a compound component and bonds external to the component. If an internal bond has a named **SS** component (i.e. **SS:[name]**) then this defines the name of that port (i.e [name]) and an external bond, connecting to this port, has the port label [name] (that is with **SS:** removed). Naturally the user may define aliases as detailed in Figure 4.2.
- 2 **SS:i\_s** : are **SS** components which define the extent of the overall system model
- 3 Causal strokes have been assigned to each bond, providing the system with flow inputs (velocities and angular velocity), and therefore effort outputs (forces and torque). These causal strokes result in integral causality for mass inertia elements. Applying effort causality to one or more bonds will result in one or more inertia elements having differential causality.
- 4 In the file `Support_lbl.txt` components each of the **SS:[i\_s]** and **SS:i\_s** have been declared. Each of these components uses the **SS** constitutive relationship (**CR**). Associated with this **CR** are two information fields: The first is associated with the effort variable and the second with the flow variable. These fields can take a number of forms, which are detailed in Table 4.4. For each of the **SS:i\_s** components the variable `internal` has been assigned to each of the effort fields and the numeric value 0 for each of the flow fields. These declarations suppress the effort output and



argument	meaning
external	denotes an input or output
internal	suppress output (not to be used with inputs)
a number	numerical input or imposed output
a symbol (e.g $m_1 \cdot g$ )	symbolic input or imposed output
unknown	used to solve algebraic loops.
zero	used to solve algebraic loops.

Table 4.4: *Table of possible SS field arguments*

impose zero velocities (i.e. the support is stationary).

## 4.5 A Bond Graph of a Two Dimensional Mass

In this section, a physical three dimensional mass is to be modelled as a two dimensional mass. As such the idea of correspondence between suspension points between the two and three dimensional masses need to be defined: This correspondence can be seen in Figure 4.4. Here a three dimensional mass with six suspension points can be seen. This has an equivalent 2-dimensional representation with three suspension points.

In isolation, each mass element has three degrees of freedom (two orthogonal linear and one angular). However, individual masses are constrained by wires and hence the point at which a wire is attached plays a critical role in the resultant dynamics of the mass. The suspension point, and its associated transformation, relates the force and velocity at the point of attachment to the force and velocity at the centre of mass. Each mass requires the position of each suspension point to be defined, and hence a mass element is not generic; each mass component is uniquely defined by its number of suspension points. However, a number of mass components have been created: A **TestMass** with two suspension points, an **UpperMass** with three suspension points and an **IntermediateMass** with four suspension points. These cover the range of mass types, as currently used by GEO 600. However, having read the rest of this section, a user should have no problems creating alternatively

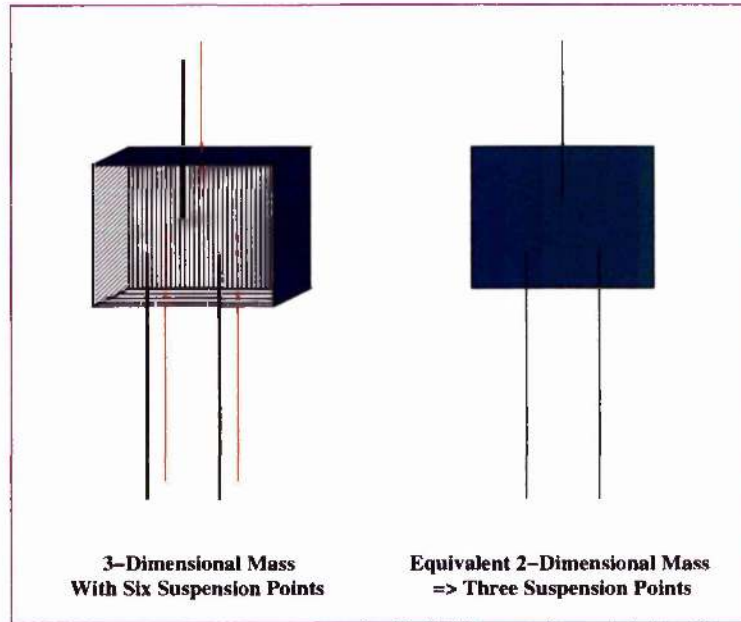


Figure 4.4: *The correspondence between the number of suspension points associated with a physical three dimensional mass and its two dimensional model*

configured mass elements. The bond graph of the upper mass is now detailed.

#### 4.5.1 The UpperMass Bond Graph Component

The bond graph of the upper mass component is shown in Figure 4.5, and as can be seen there are three vector **SS:[named]** components (**SS:[up]**, **SS:[dl]**, **SS:[dr]**)<sup>5</sup>, each vector is comprised of three bonds, one for each of the X,  $\Theta$  and Y coordinates ([1,2,3] respectively). Each vector bond carries the co-variables relating to individual suspension points (upper, lower left and lower right). These connect to appropriate ports of the subcomponent **UpperMassSusp**. This subcomponent transforms the effort and flow variables at a suspension point to effort and flow at the centre of mass. A full description of this subcomponent is detailed in the next subsection. A further four ports exist at the component **UpperMassSusp**, three of these,  $[x_{cm}, a_{cm}, y_{cm}]$ , are the centre of mass ports, one each for the X,  $\Theta$  and Y coordinates. These are in turn connected to individual **1** junctions. Remember a **1** junction

<sup>5</sup>vector **SS** ports are explained in Figure 4.2

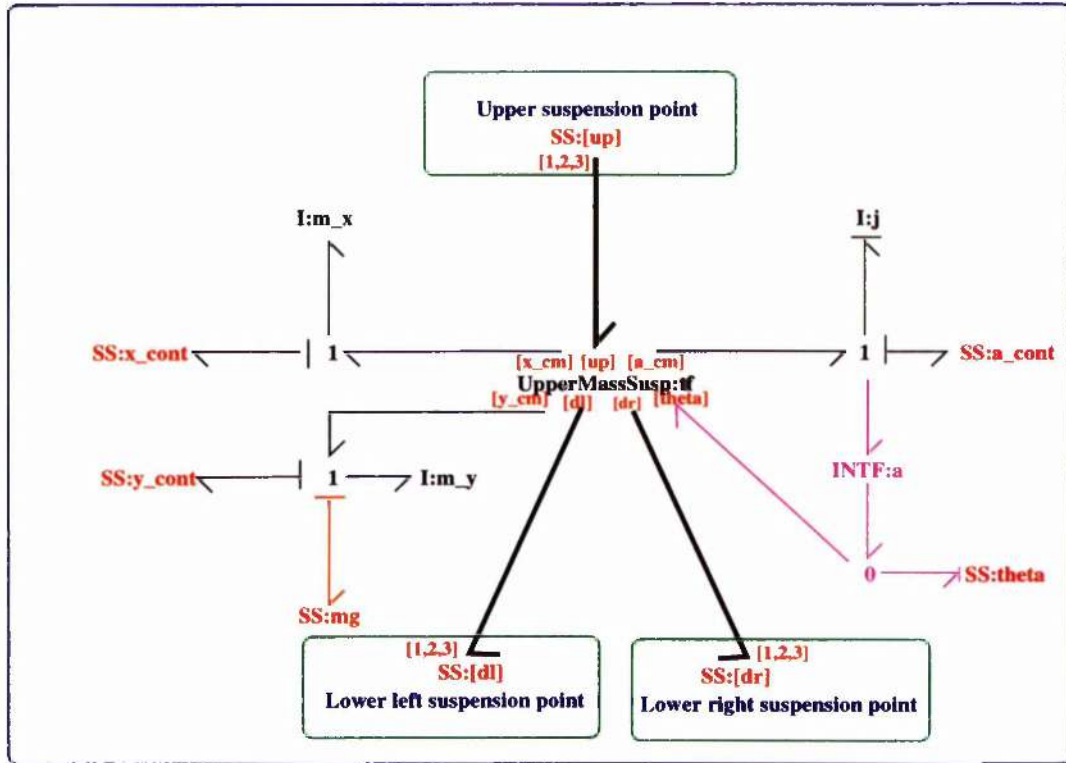


Figure 4.5: The bond graph component *UpperMass*

has a common flow input and  $\sum \text{efforts} = 0$ . At each junction an effort store (**I** component) (**I:m<sub>x</sub>**, **I:j**, **I:m<sub>y</sub>**), with integral causality, provides a state to the overall system model, and imposes the appropriate flow variable on their respective **1** junctions.

These inertia components (**I**) correspond to the three degrees of freedom for a two dimensional mass. Note that as part of a system model a mass component may be constrained such that one or more of the **I** components has differential causality, i.e. that energy store is no longer independent, and hence there is a corresponding reduction in the number of degrees of freedom possessed by the mass. However the same mass component is used no matter the causal arrangement: Thus exemplifying the re-usability of bond graph components under differing causal configurations.

In addition to the inertia components (**I**), each of the **1** junctions is augmented with a control input (**SS:cont<sub>x</sub>** etc) which provides an interface for control forces (bond graph control

synthesis is discussed in Chapters 6 and 7). A further bond (SS:mg) appears at the **1** junction associated with the Y coordinate, this source/sensor component imposes the force due to the gravitational attraction of the Earth on the mass. Finally there is an **INTF** component (Section 3.7) attached to the **1** junction with the conjoined **I:j** component and is used to integrate the flow variable without imposing a corresponding “back” effort. Here the **INTF:a** component integrates the angular velocity to give the angle through which the mass has rotated. This variable is required for use within the effort modulating transformers found within the subcomponent **CoordTrans** (see Section 4.5.3) and to this end the **[theta]** port is provided at the **UpperMassSusp** component (as will be seen **UpperMassSusp** contains three instances of **CoordTrans**). The **UpperMassSusp** component will now be discussed.

#### 4.5.2 The Bond Graph Component UpperMassSusp

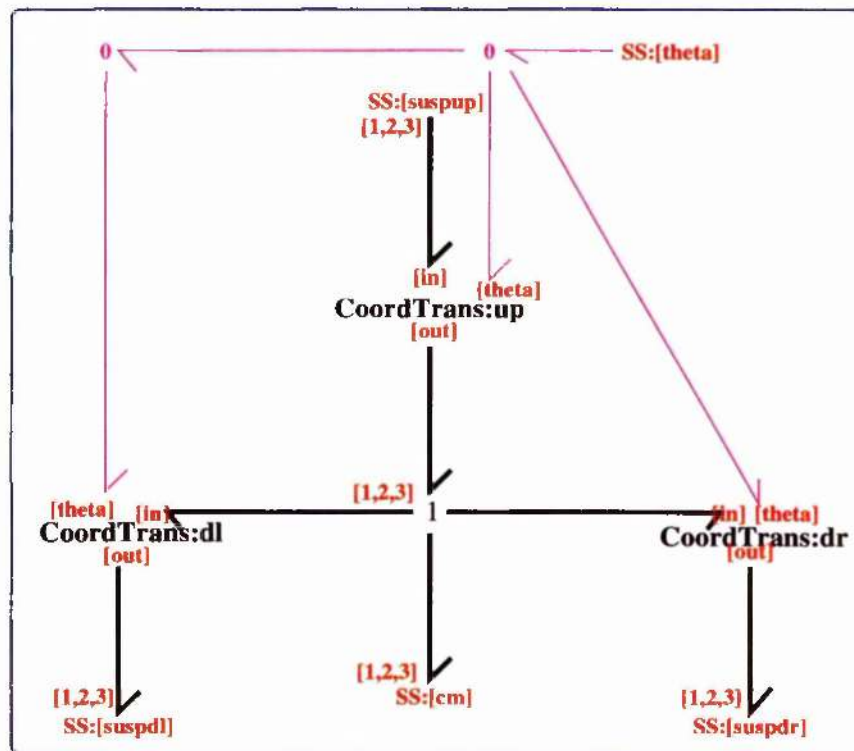


Figure 4.6: The bond graph component **UpperMassSusp**

The bond graph **UpperMassSusp** can be seen in Figure 4.6. Most of the structure of this



bond graph should now be familiar to the user. Apart from the new component **CoordTrans** the only other new element is the **MTT vector junction 1**<sup>6</sup>, denoted by the label [1,2,3]. Here four vector bonds converge on the **1** junction. The label [1,2,3] is a marker for MTT and signifies that each vector bond, converging on the junction, consists of three bonds. These three bonds are just those associated with the X,  $\theta$  and Y coordinates ([1,2,3] respectively). This vector junction with its associated vector bonds is equivalent to the bond graph structure of Figure 4.7, but is clearly easier to comprehend.

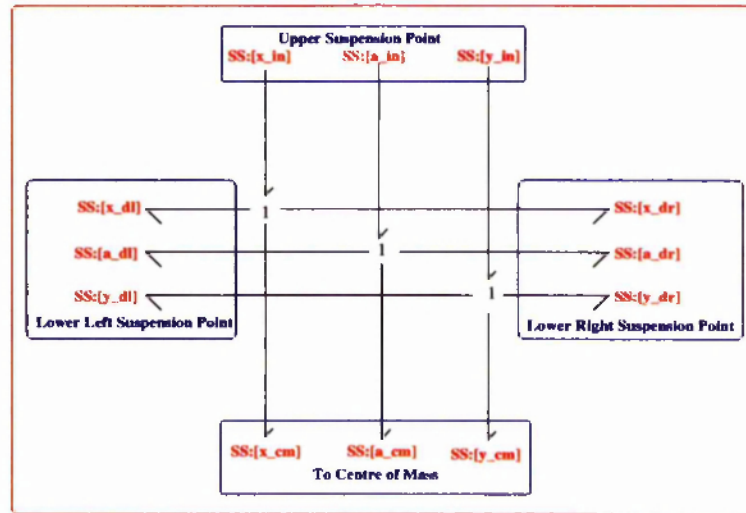


Figure 4.7: *The equivalent structure of a vector 1 junction as used within component UpperMassSusp*

Turning to the component **CoordTrans**, this is the key to the successful implementation of a mass element constrained by suspension wires. The position of a wire suspension point, relative to the centre of mass, is fundamental to the resultant dynamics of the system model. Fortunately, because the model is modular, it is not necessary to determine how the overall topology of the system effects the dynamics of an individual mass element. All that is necessary is to determine the correct transformations between co-energy variables at a suspension point to co-energy variables at the centre of mass. Having created a bond graph of this transformation it is simple, as will be demonstrated, to include as many suspension points as is necessary to model the system correctly. The correct transformation, and bond

<sup>6</sup>Naturally a similar vector **0** junction is permitted.

graph implementation, now follows.

### 4.5.3 The Bond Graph Component CoordTrans

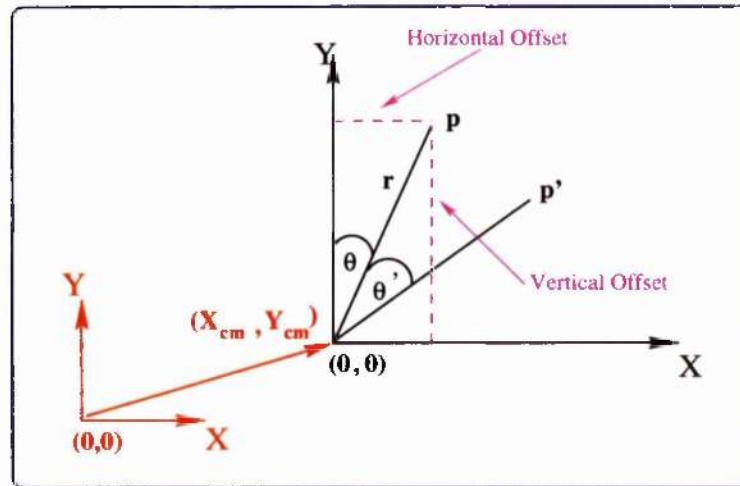


Figure 4.8: *Definition of coordinate system*

The fundamental elements of transformers and transformations, within the context of bond graphs, were discussed in Section 3.6.1. In this section it will be shown how the geometric transformation, required to correctly model 2-dimensional mechanical systems, is generated in the bond graph domain.

Referring to Figure 4.8 let  $p$  represent the position of a suspension point. Then  $h$  and  $v$ , the horizontal and vertical offsets of the suspension point respectively, are given by:-

$$\begin{aligned} h &= r \sin(\theta) \\ v &= r \cos(\theta) \end{aligned} \quad (4.1)$$

Now if the point  $p$  is rotated through an angle  $\theta'$  about the centre of mass the new position

is given by:-

$$\begin{aligned}x_p' &= r \sin(\theta + \theta') = h \cos(\theta') + v \sin(\theta') \\y_p' &= r \cos(\theta + \theta') = -h \sin(\theta') + v \cos(\theta')\end{aligned}\quad (4.2)$$

Combining this motion with pure translational motion of the centre of mass gives:-

$$\begin{aligned}x_p'' &= x_{cm} + h \cos(\theta') + v \sin(\theta') \\y_p'' &= y_{cm} - h \sin(\theta') + v \cos(\theta')\end{aligned}\quad (4.3)$$

where  $x_p''$  and  $y_p''$  are the points  $x_p'$  and  $y_p'$  of the moving frame expressed in the absolute frame,  $x_{cm}$  and  $y_{cm}$  are the  $x$  and  $y$  coordinates of the centre of mass with respect to the absolute coordinate system. Taking the time derivative of these equations results in a pair of equations relating the angular and translational velocities, at a point in a rigid body, to the angular and translational velocities at its centre of mass, i.e (dropping ' from  $\theta'$ ) the  $x$  and  $y$  components of velocity, at the centre of mass are:

$$\begin{aligned}\dot{x}_{cm} &= \dot{x}_p + h \sin(\theta)\dot{\theta} - v \cos(\theta)\dot{\theta} \\ \dot{y}_{cm} &= \dot{y}_p + h \cos(\theta)\dot{\theta} + v \sin(\theta)\dot{\theta}\end{aligned}\quad (4.4)$$

This transformation is embodied in the bond graph of Figure 4.9.

Here the component **EMTF** (Effort Modulating Transformer) is an elemental MTT component, where "Effort Modulation" relates to the bond marked [mod]. This bond carries the modulating variable (in this circumstance  $\theta$  - the angle through which the mass has rotated) as the effort variable. Each **EMTF** has a constitutive relationship that relates the output flow to the input flow variable in one of two ways, either

$$f_{out} = l \cos(\theta) f_{in}$$

or

$$f_{out} = l \sin(\theta) f_{in}\quad (4.5)$$

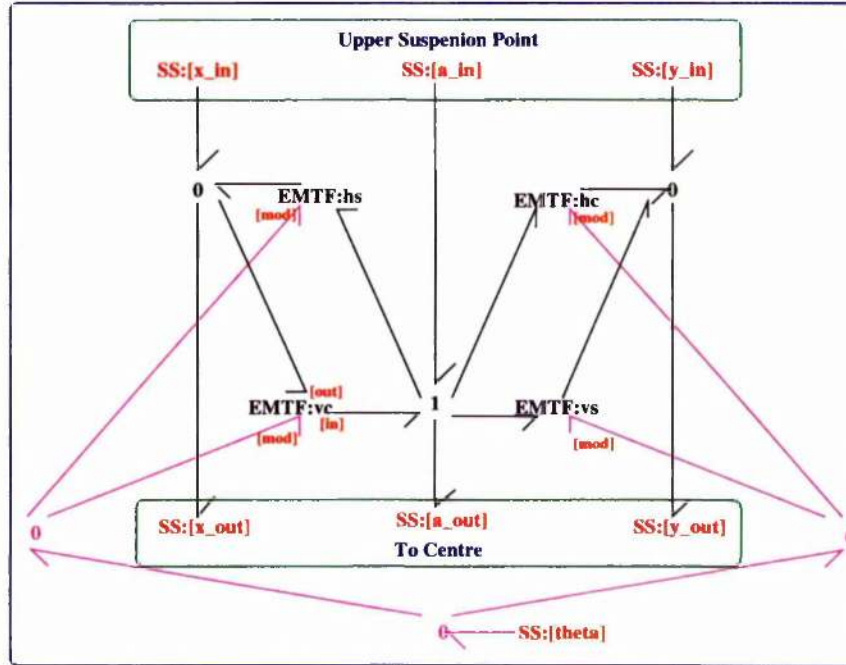


Figure 4.9: Transformation relating suspension point velocities to centre of mass velocities.

Points to note are:

- For each **EMTF** component the choice of **CR** is indicated by the component name (e.g **EMTF:hs**), where the first letter denotes whether the  $h$  (horizontal offset) or  $v$  (vertical offset) should be substituted for  $l$ , and the second letter indicates whether the sin or cosine variant of **CR** is used.
- In each case the input flow ( $f_{in}$ ) is  $\dot{\theta}$  (angular velocity) the sign of which is evident from the direction of the bond.
- The corresponding relationship between the effort input and output variables is simply the transpose of the matrix relating the input and output flow variables (conservation of energy and transformation properties of junction structures – Section 3.6.1).
- The signs of the parameters  $h$  and  $v$  are precisely those given for a Cartesian coordinate system with the centre of mass as the origin

It is possible to use MTT to check the consistency of components. This will be illustrated



by producing the ordinary differential equations for the **CoordTrans**. The first task is to check that MTT is able to create the `CoordTrans_abg.m` by issuing the command:

```
prompt> mtt -I CoordTrans abg m
```

This creates an m file of the `ComponentName_abg.fig`. The `-I` switch prompts MTT to return extra information. As MTT carries out the command, information will be displayed, and should an error occur an appropriate error message will appear. The user will then need to check the acausal bond graph and/or the `lbl` file for errors. Assuming all is okay, execute the command

```
prompt> mtt -I CoordTrans cbg m
```

MTT will attempt to complete causality, by default integral causality is imposed and naturally if MTT is unable to complete causality it will return an error message. Causal conflicts should be resolved as detailed in Section 3.6. (Since **CoordTrans** has no energy storage ports, causal strokes, appropriate for force inputs at the centre of mass, were applied.)

Having successfully completed causality, issue the command

```
prompt> mtt -I CoordTrans struc tex (or view or txt)
```

this generates the *structure* file for the component **CoordTrans** where `tex`, `txt` and `view` will produce files of the form `CoordTrans_struc.tex`, `CoordTrans_struc.txt` or `CoordTrans_struc.dvi` respectively (again `view` will produce a new window containing the `CoordTrans_struc.dvi` file)<sup>7</sup>. The two tables that form the **CoordTrans** structure file can be seen in Table 4.5. Generally a structure file consists of four tables, listing the system inputs, outputs, states and non states respectively (remember non states are produced when a bond graph energy port has derivative causality). However since **CoordTrans** has no states or non states these tables are redundant. These tables are not the exact files as generated by MTT, the names in brackets have been added for clarity.

---

<sup>7</sup>Both of the `ode` and `struc` for the **CoordTrans** files are MTT generated `tex` files

List of inputs for system CoordTrans			
	Component	System	Repetition
1	x_in (Velocity)	CoordTrans	1
2	a_in (Angular Velocity)	CoordTrans	1
3	y_in (Velocity)	CoordTrans	1
4	x_out (Force)	CoordTrans	1
5	a_out (Torque)	CoordTrans	1
6	y_out (Force)	CoordTrans	1
7	theta (Modulation Variable)	CoordTrans	1

List of outputs for system CoordTrans			
	Component	System	Repetition
1	x_in (Force)	CoordTrans	1
2	a_in (Torque)	CoordTrans	1
3	y_in (Force)	CoordTrans	1
4	x_out (Velocity)	CoordTrans	1
5	a_out (Angular Velocity)	CoordTrans	1
6	y_out (Velocity)	CoordTrans	1
7	theta (0)	CoordTrans	1

Table 4.5: The structure files for the **CoordTrans** component

Now create the ordinary differential equations by issuing the command:

```
prompt> mtt -I CoordTrans ode view
```

this will create a dvi file `CoordTrans_ode.dvi` and should automatically produce a new window displaying the file containing the ordinary differential equations (ode's). The ode's for the **CoordTrans** are shown in Equation 4.6. Note that because there are no dynamics, these are algebraic equations.

$$\begin{aligned}
y_1 &= u_4 \\
y_2 &= (hu_6 - u_4v) \cos(u_7) + u_5 + (hu_4 + u_6v) \sin(u_7) \\
y_3 &= u_6 \\
y_4 &= \sin(u_7)hu_2 + u_1 - \cos(u_7)u_2v \\
y_5 &= u_2 \\
y_6 &= \sin(u_7)u_2v + u_3 + \cos(u_7)hu_2 \\
y_7 &= 0
\end{aligned} \tag{4.6}$$

Notice, with reference to Equation 4.6 and Table 4.5, that velocity outputs  $y_4$  and  $y_6$  correspond to the Equations of 4.4 and that  $y_2$  (the torque) has, due to kinematics and energy conservation, been automatically generated [43]. Hence, although the coordinate transformation was developed via the transformation of input and output velocities, the bond graph methodology correctly generates the force equations (see Section 3.6.1). Moreover, it should be appreciated that the causality, as used to generate the odes of Equation 4.4, is not unique. Yet the same bond graph component is used independent of the imposed causality.

This covers the components that make up the **UpperMass** component. The only difference between the **UpperMass**, **TestMass** and **IntermediateMass** components are the number of suspension points. Moreover, following this introduction, a user should be in the position to produce masses with any number of suspension points.

Having created the bond graph component **UpperMass**, the next stage is to create the label file, Table 4.6:

```
prompt> mtt UpperMas lbl txt
```

Again **MTT** automatically creates the text in the `typewriter` typeset and the *italicised* text must be supplied by the user. Points to note are:

- Vector ports have been declared (as explained in Figure 4.2 – port alias declarations).

```

%% Label file for system UpperMass (UpperMass_lbl.txt)
%SUMMARY UpperMass
%DESCRIPTION <Detailed description here>
% *****
% %% Version control history
% *****
% %% $Id$
% %% $Log$
% *****
% Port aliases
%ALIAS   up      up_1,up_2,up_3
%ALIAS   dl      dl_1,dl_2,dl_3
%ALIAS   dr      dr_1,dr_2,dr_3
%ALIAS   in      up_1,up_2,up_3
%ALIAS   out     dl_1,dl_2,dl_3,dr_1,dr_2,dr_3
% Argument aliases
%ALIAS   $1      m_1
%ALIAS   $2      j_1
%ALIAS   $3      g
%ALIAS   $4      hu_1
%ALIAS   $5      hl_1
%ALIAS   $6      hr_1
%ALIAS   $7      vu_1
%ALIAS   $8      vl_1
%ALIAS   $9      vr_1
%% Each line should be of one of the following forms:
%          a comment (ie starting with %)
%          component-name      cr_name arg1,arg2,..argn
%          blank
% ---- Component labels ----
% Component type I
%          j      lin      flow,j_1
%          m_x    lin      flow,m_1
%          m_y    lin      flow,m_1
% Component type INTF
%          x
%          a
%          y
% Component type SS
%          [up]   SS      external,external
%          [dl]   SS      external,external
%          [dr]   SS      external,external
%          x_cont SS      external,internal
%          a_cont SS      external,internal
%          y_cont SS      external,internal
%          x_disp SS      external,0
%          theta  SS      external,0
%          y_disp SS      external,0
%          mg     SS      m_1*g,0
% Component type UpperMassSusp
%          tf     none    hu_1;hl_1;hr_1;vu_1;vl_1;vr_1

```

Table 4.6: *UpperMass\_lbl.txt* file



- Notice, the component declarations are similar to those of `Double_lbl.txt` (Page 53). However here the constitutive relationship “lin” is declared<sup>8</sup>. This is a standard linear constitutive relationship, predefined within MTT, requiring two arguments. The first argument gives the name of the input variable and the second a named gain. Examples are given below.

	arguments	equation	example
I	flow,m	state=m×flow	P=m v; momentum = mass×velocity
C	state,k	effort=k×state	F=k x; force = k×displacement
R	effort,1/r	flow = $\frac{1}{r}$ × effort	i=v/r; current= voltage/resistance

- Notice that arguments needed by a **CR** (i.e lin here) are separated by “;” whereas arguments passed to another component (i.e where the constitutive relationship field is declared as none) are separated by “,”.
- Components, such as **INTF**, where no **CR** is declared, default to a linear **CR** with gain of 1.
- **SS** components all use the **SS** constitutive relationship. The possible arguments for the two information fields were discussed in Section 4.4 and displayed in Table 4.4 (Page 56).
- Argument aliasing is utilised. This allows parameters, contained within a higher level bond graph `lbl` file, to be passed to a lower level bond graph. For example, in the `Double_lbl.txt` file the line (associated with the **UpperMass**) appears

```

mass_1      none      m_1; j_1; g...etc
           field 1    field 2

```

where the second field contains a list of the parameters needed by component **UpperMass**. These are passed to **UpperMass** in the order shown, that is `m_1` is associated with `$1`, `j_1` with `$2`, etc. Although each `$number` declaration, of `UpperMass_lbl.txt` has the same name as those that appear in the higher level `Double_lbl.txt` file, this is not necessary. In fact these are just dummy arguments

<sup>8</sup>NB: As of writing this thesis the method by which a user declares constitutive relationships is undergoing a major rewrite. Therefore one should check the user manual.

which correspond to the same named parameters found later within the `Upper-Mass_1b1.txt` file. These dummy arguments are replaced by the named parameters of the higher level component. Hence multiple use of the same component is permitted within a bond graph, whereby appropriate named parameters are passed from the higher level bond graph to the subcomponent. Each subcomponent will then have a unique set of parameters.

The wire component will now be discussed.

## 4.6 Construction of a Wire Bond Graph

The **Wire** component is a complex system and as such only a simplified wire model will be presented here. Chapter 5 gives a full description of the **Wire** component's development. The **Wire** component presented here is deemed to be a semi-rigid wire. It is modelled to have linear extension but does not include bending dynamics or transverse (violin) modes.

### 4.6.1 Modeling a Semi-Rigid Wire Component

The important characteristics of a wire suspension, in terms of restoring forces, are the angle the wire makes with the vertical, and the change in it's length. The first task of any bond graph model should be to determine these.

First consider two points (A and B) as shown in Figure 4.10. Their respective positions are given by the following set of equations:-

$$\begin{aligned} X_A &= \frac{l}{2} \sin(\theta) + X_{cm} \\ Y_A &= \frac{l}{2} \cos(\theta) + Y_{cm} \\ X_B &= -\frac{l}{2} \sin(\theta) + X_{cm} \\ Y_B &= -\frac{l}{2} \cos(\theta) + Y_{cm} \end{aligned} \tag{4.7}$$

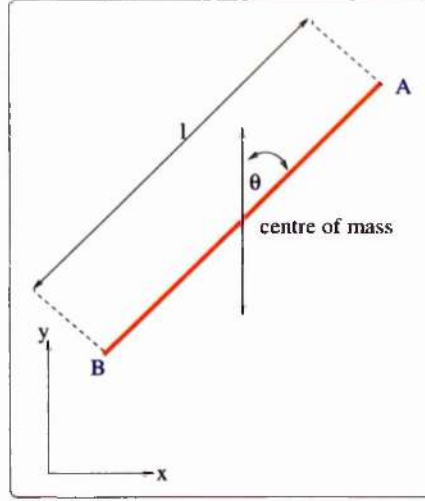


Figure 4.10: *Geometry of a wire*

and hence their velocities are given by:-

$$\dot{X}_A = \frac{1}{2}[l \cos(\theta)\dot{\theta} + \dot{l} \sin(\theta)] + \dot{X}_{cm} \quad (4.8)$$

$$\dot{Y}_A = -\frac{1}{2}[l \sin(\theta)\dot{\theta} - \dot{l} \cos(\theta)] + \dot{Y}_{cm}$$

$$\dot{X}_B = -\frac{1}{2}[l \cos(\theta)\dot{\theta} + \dot{l} \sin(\theta)] + \dot{X}_{cm}$$

$$\dot{Y}_B = \frac{1}{2}[l \sin(\theta)\dot{\theta} - \dot{l} \cos(\theta)] + \dot{Y}_{cm}$$

which can be simultaneously solved for  $\dot{\theta}$ , the time rate change of the angle, and  $\dot{l}$ , the time rate change of the wire length.

$$\dot{\theta} = \frac{\Delta \dot{x} \cos(\theta) - \Delta \dot{y} \sin(\theta)}{l} \quad (4.9)$$

$$\dot{l} = \Delta \dot{x} \cos(\theta) + \Delta \dot{y} \sin(\theta) \quad (4.10)$$

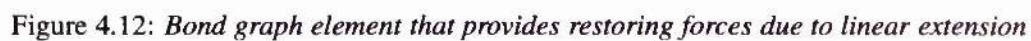
Where  $\Delta \dot{x} = \dot{x}_A - \dot{x}_B$  and  $\Delta \dot{y} = \dot{y}_A - \dot{y}_B$ .

Hence, via integration of these variables, the angle and wire extension are generated. The (black) bonds contained within the dashed border of Figure 4.11 form the bond graph realisation of Equations 4.9.

Having created this component, various subcomponents can be embedded within it to generate any necessary dynamics. These may include violin modes, restoring forces due to



#### 4.6.2 The bond graph component LinExt



71



```

%% Label file for system LinExt (LinExt_lbl.txt)=\\
%SUMMARY LinExt simply models restoring forces=\\
%SUMMARY due to linear extension=\\
%DESCRIPTION <Detailed description here>=\\

% #####
% %% Version control history=\\
% #####
% %% $Id$=\\
% %% $Log$=\\
% #####

% Port aliases
%ALIAS      in|a_a      angular_a
%ALIAS      out|a_b     angular_b
%ALIAS      x           linear

% Argument aliases
%ALIAS      $1          k
%% Each line should be of one of the following forms:
%      a comment (ie starting with %)
%      component-name   cr_name arg1,arg2,..argn
%      blank

% ---- Component labels ----

% Component type C
%           c          lin          state,k

% Component type SS
%           [angular_a]  SS          external,external
%           [angular_b]  SS          external,external
%           [linear]     SS          external,external
%           zero         SS          0,internal

```

Table 4.7: *LinExt\_lbl.txt File*

three bonds ensures that the angle of rotation of the mass and wire at the attachment points are not fixed, i.e it is as if the wire's ends were attached by frictionless hinges. This is achieved via the named **SS:zero** source/sensor component which imposes zero force (see Table 4.7) on the **0** junction.

Restoring forces, due to linear extension of the wire, are provided by the named flow energy store **C:k**. As stated in Chapter 3, a **C** component has a constituent relationship that relates the effort variable to  $\phi(q)$  (the general displacement) where  $q = \int f dt$ . In the linear case  $e = \frac{1}{C}q$ . Thus, for a wire whose extension obeys Hooke's law the **Wire** and **LinExt** components model the appropriate system dynamics.

List of inputs for system Wire			
	Component	System	Repetition
1	a_in	Wire	1
2	a_out	Wire	1
3	x_in	Wire	1
4	x_out	Wire	1
5	y_in	Wire	1
6	y_out	Wire	1

List of outputs for system Wire			
	Component	System	Repetition
1	a_in	Wire	1
2	a_out	Wire	1
3	x_in	Wire	1
4	x_out	Wire	1
5	y_in	Wire	1
6	y_out	Wire	1

List of states for system Wire			
	Component	System	Repetition
1	c	Wire_wire	1
2	mtt3	Wire_theta	1

Table 4.8: *The structure file for the Wire bond graph. Inputs are flows and outputs are efforts*

### 4.6.3 Using MTT to check the consistency of this bond graph wire component

The structure file for the above detailed wire component can be viewed in Table 4.8. Here inputs are flows (linear and angular velocities) and the outputs are efforts (forces and torques).

By executing the MTT command:

```
prompt> mtt -I Wire ode view
```

the ordinary differential equations, Equations 4.11 and 4.12, will be created for this wire model, which simply models the restoring forces due to linear extension.

$$\begin{aligned}\dot{x}_1 &= (u_3 - u_4) \sin(x_2) + (u_5 - u_6) \cos(x_2) \\ \dot{x}_2 &= \frac{((u_3 - u_4) \cos(x_2) - (u_5 - u_6) \sin(x_2))}{l}\end{aligned}\tag{4.11}$$

$$\begin{aligned}y_1 &= 0 \\ y_2 &= 0 \\ y_3 &= \sin(x_2)kx_1 \\ y_4 &= \sin(x_2)kx_1 \\ y_5 &= \cos(x_2)kx_1 \\ y_6 &= \cos(x_2)kx_1\end{aligned}\tag{4.12}$$

Here,  $x_i$  are the system states ( $x_1$  is the wire extension and  $x_2$  the angle the wire makes with the vertical),  $u_i$  are the inputs, and  $y_i$  are the outputs, as detailed in the **Wire** structure file of Table 4.8. As can be seen  $\dot{x}_1$  and  $\dot{x}_2$  suitably replicate the Equations of 4.9. Moreover, in accordance with the principles expounded in Section 4.5.3 and Section 3.6.1, the restoring forces are exactly those one would expect for this “simple” wire model. Also, the two torques (outputs  $y_1$  and  $y_2$ ) are, as expected, equal to zero (remember, the **SS:zero** component, of component **LinExt** ensures this is the case).

#### 4.6.4 The bond graph component **TwoWires**

For dual suspensions, the bond graph **Twowires** (Figure 4.13) is used. Here two instances of a single wire are combined to produce a single two wire component. This illustrates the construction of a simple hierarchical bond graph component. The label file (**TwoWires\_lbl.txt**) for this component can be seen in Table 4.9. Notice that in this lbl file port alias statements **in** and **out** have been declared (a description of port aliasing is given in Figure 4.2). This results in a two port component with six vector bonds at each port. Hence, when used as a subcomponent of a hierarchical bond graph, such as the double pendulum model of Figure 4.1 (Page 51), a simple **[in]/[out]** declaration can be used to conjoin this component with others.

This then completes all the components needed to model a two dimensional double pendulum. The next section draws all these components together, illustrates some more MTT usage and compares the natural mode frequencies of a physical double pendulum with those generated by equivalent bond graph and Matlab models.

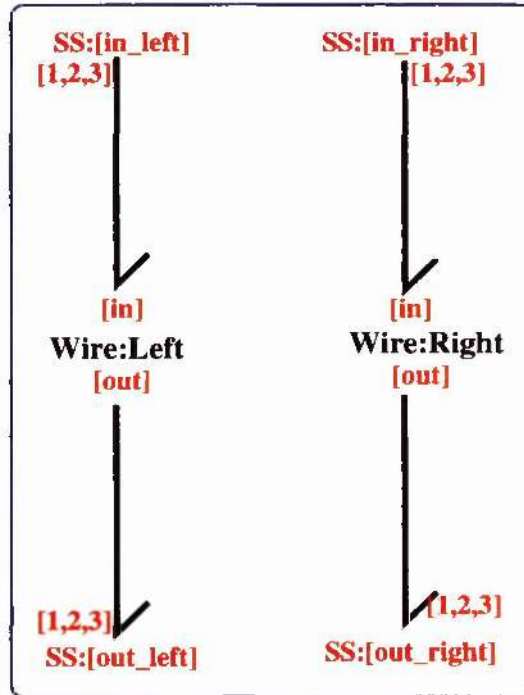


Figure 4.13: *The bond graph component TwoWires*

## 4.7 MTT and the Double Pendulum

MTT has many features, such as the ability to carry out model simulation, sensitivity analysis (experimental) and generate alternative system representations such as transfer functions and state matrices<sup>9</sup>. The procedure for generating state matrices, using MTT, will now be demonstrated.

Having checked that MTT can generate the Double\_cbg.m file (i.e. the causally complete bond graph exists) the structure file for the double pendulum can be created using the

<sup>9</sup>All these and other features are documented within the MTT manual

```

%% Label file for system TwoWires (TwoWires_lbl.txt)
%SUMMARY TwoWires
%DESCRIPTION <Detailed description here>
% *****
% %% Version control history
% *****
% %% $Id$
% %% $Log$
% *****
% Port aliases
%ALIAS in in_left_1,in_left_2,in_left_3,in_right_1,in_right_2,in_right_3
%ALIAS out out_left_1,out_left_2,out_left_3,out_right_1,out_right_2,\
out_right_3
% Component aliases
% Argument aliases
%ALIAS $1 1
%ALIAS $2 k
%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% component-name cr_name arg1,arg2,..argn
% blank
% ---- Component labels ----
% Component type SS
[in_left] SS external,external
[in_right] SS external,external
[out_left] SS external,external
[out_right] SS external,external
% Component type Wire
Left none 1;k
Right none 1;k

```

Table 4.9: *TwoWires\_lbl.txt* file

command:

```
prompt> mtt -I Double struc view
```

Again, the structure file gives information about the inputs, outputs, system states and non states, and those corresponding to the double pendulum can be seen in Table 4.10.

The ordinary differential equations (ode) can be created and viewed by issuing the command:-

```
prompt> mtt -I Double ode view
```

This will produce the system ode's. These are symbolic equations, although it is possible to supply simplifying equations/numerical values, such as defining trigonometric identities, via the `SystemName_subs.r` file, (see MTT documentation)<sup>10</sup>.

The odes are used to determine the steady state of the system. The steady state is found by setting the state derivatives to zero and solving the resulting simultaneous equations for the states. Having solved these equations the results are passed to MTT via the *sspar* (steady state parameter) file. A prototype file is created via the command:

```
prompt> mtt Double sspar r
```

Where “r” signifies that this file is a reduce language file. The *sspar* (*steady state parameter*) file for Double (`Double_sspar.r`) can be seen in Table 4.11.

Initially all values are set to zero and the user should supply the relevant details. For this double pendulum the steady state values have been supplied corresponding to the extension of the wires (MTTX13, MTT15 and MTT17). Where MTTX13 represents state  $x_{13}$  etc.

---

<sup>10</sup>Basically the `SystemName_sub.r` file is a Reduce file that is called by MTT whilst executing transformations, such as ordinary differential equations, and can be used to change the expressions describing a system. These effect all system transformations. Alternatively, if the same expressions are placed in the file `SystemName_simp.r` only the representations in the  $\text{\LaTeX}$  format are changed.

List of inputs for system Double			
	Component	System	Repetition
1	act_1	Double	1
2	act_2	Double	1
3	act_3	Double	1
4	theta	Double	1

List of outputs for system Double			
	Component	System	Repetition
1	act_1	Double	1
2	act_2	Double	1
3	act_3	Double	1
4	angle	Double.mass_1	1

List of states for system Double			
	Component	System	Repetition
1	j	Double.mass_2	1
2	m_x	Double.mass_2	1
3	m_y	Double.mass_2	1
4	mtt3	Double.mass_2.a	1
5	c	Double.suspension_2_Left_wire	1
6	mtt3	Double.suspension_2_Left_theta	1
7	c	Double.suspension_2_Right_wire	1
8	mtt3	Double.suspension_2_Right_theta	1
9	j	Double.mass_1	1
10	m_x	Double.mass_1	1
11	m_y	Double.mass_1	1
12	mtt3	Double.mass_1.a	1
13	c	Double.suspension_1_wire	1
14	mtt3	Double.suspension_1_theta	1

Table 4.10: The structure table for the double pendulum hierarchical bond graph



```

% Steady-state parameter file (Double_sspar.r)
% Generated by MTT at Mon Oct  4 12:38:26 BST 1999

% =====
% % Version control history
% =====
% % $Id$
% % $Log$
% =====

% Steady-state states
MTTX1 := 0; % Double_mass_1 (j)
MTTX2 := 0; % Double_mass_1 (m_x)
MTTX3 := 0; % Double_mass_1 (m_y)
MTTX4 := 0; % Double_mass_1_ang (mtt3)
MTTX5 := 0; % Double_mass_1_x (mtt3)
MTTX6 := 0; % Double_mass_1_y (mtt3)
MTTX7 := 0; % Double_mass_2 (j)
MTTX8 := 0; % Double_mass_2 (m_x)
MTTX9 := 0; % Double_mass_2 (m_y)
MTTX10 := 0; % Double_mass_2_ang (mtt3)
MTTX11 := 0; % Double_mass_2_x (mtt3)
MTTX12 := 0; % Double_mass_2_y (mtt3)
MTTX13 := g*(m_1+m_2)/(cos(MTTX14)*k_1);%1_wire(c)
MTTX14 := 0; % Double_suspension_1_theta (mtt3);
MTTX15 := (1/2)*g*m_2/(cos(MTTX16)*k_2);%2_Left_wire(c)
MTTX16 := 0; % Double_suspension_2_Left_theta (mtt3)
MTTX17 := (1/2)*g*m_2/(cos(MTTX18)*k_2);%2_Right_wire(c)
MTTX18 := 0; % Double_suspension_2_Right_theta (mtt3)

% Steady-state inputs
MTTU1 := 0; % Double_mass_2 (x_cont)
MTTU2 := 0; % Double_mass_2 (a_cont)
MTTU3 := 0; % Double_mass_2 (y_cont)
MTTU4 := 0; % Double_mass_1 (x_cont)
MTTU5 := 0; % Double_mass_1 (a_cont)
MTTU6 := 0; % Double_mass_1 (y_cont)
MTTU7 := 0; % Double_mtt2 (x_s)
MTTU8 := 0; % Double_mtt2 (a_s)
MTTU9 := 0; % Double_mtt2 (y_s)
;;END;

```

Table 4.11: *Double\_sspar.r* file: The steady state parameter file for system *Double*

The forces corresponding to wire extension balance the weights of the masses. The relevant equations relating these forces are shown in Equation 4.13. These were taken from the set of system ode's (for clarity others are not shown here as they do not effect the steady state).

$$\dot{x}_3 = \cos(x_6)k_2x_5 + \cos(x_8)k_2x_7 - gm_2 \quad (4.13)$$

$$\dot{x}_{11} = \cos(x_{14})k_1x_{13} - \cos(x_6)k_2x_5 - \cos(x_8)k_2x_7 - gm_1 + u_3$$



Solving these for  $\dot{x}_3 = \dot{x}_{11} = 0$  gives the equilibrium extension of each wire, which can then be appended to the steady state parameter file (`Double_sspar_r`). Here it has been assumed that wires supporting masses are symmetric, and hence the weight of the supported mass is evenly distributed between the wires. It should be appreciated that finding the system's steady state is fundamental to generating the correct state matrices. This is because pendulum models, due to geometric transformations, are inherently non linear. Once the steady state has been determined a linearisation, as discussed in Section 3.9, can be performed. This linearisation will automatically be executed by MTT when transforming from the ordinary differential equations to other representations such as state space equations or transfer function. However prior to executing these transformations the supplied steady state values can be checked by executing the MTT command:

```
prompt> mtt -I Double ss view
```

“ss” denotes steady state and again view will present the file of the `Double_ss.dvi` in a separate window and thus the state derivatives can be checked to see whether they are exactly zero. For the `sspar` file above this is the case, and hence state matrices, transfer functions etc may now safely be produced.

```
prompt> mtt -I Double sm m
```

and

```
prompt> mtt -I Double tf m
```

produce state matrices and transfer functions respectively in the `m` file format (Matlab/Ocatve)<sup>11</sup> [47].

However before moving to Matlab or Octave a numerical parameter (`numpar`) file should be created.

---

<sup>11</sup>Octave is a high level computer language , primarily intended for numerical computations and uses Gnuplot for graphics. It has a control toolbox and the syntax is mostly compatible with Matlab

```
prompt> mtt -I Double numpar txt
```

This file contains a list of all the parameters used within the model. Initially all values are set to 1.0 and should be edited to give appropriate numerical values. From this file an `m` file is created by replacing `txt` by `m` in the previous command. The `numpar.m` file for Double is shown in Table 4.13.

These are the appropriate parameters for the double pendulum as used in the 10m prototype gravitational wave detector [48]. Hence, should a user wish to use the predefined double pendulum model, with a different set of parameters, then all that is required is to edit the `Double_numpar.txt` file and then re-issue the command.

```
prompt> mtt -I Double numpar m
```

## 4.8 Comparing Double Pendulum Models with the Physical System

In this section, the mode frequencies of a real physical multi-stage pendulum will be compared with equivalent bond graph and Matlab models.

The resonant mode frequencies of a real physical double pendulum system are presented in Table 4.12. These measured values are compared with those generated by the equivalent bond graph and Matlab models. The Matlab model was designed by S Killbourn [48]. The experimental error for each mode is reflected in the number of figures quoted after the decimal point [48]. The various defining parameters are detailed in the `numpar` file of Table 4.13 (Page 83).

## 4.9 Conclusions

In this chapter a library of bond graph components, for the construction of 2-dimensional multi-stage pendula, were presented. A hierarchical double pendulum bond graph model

Mode	Bond Graph	Expt	Matlab
Longitudinal	0.703	0.700	0.709
Longitudinal	1.913	1.86	1.92
Tilt	0.83	0.82	0.856
Tilt	3.18	3.24	3.15
Vertical	10.7	10.2	10.8
Vertical	27.3	28.2	29.3

Table 4.12: *Table of eigenvalues for prototype double pendulum*

was then produced and the predicted mode frequencies compared with both the equivalent real physical system and a Matlab model. The predicted frequencies were consistent with those of the physical system and as such alternative multi-stage pendulum systems can be modelled with confidence.

```

# --octave-- Put Emacs into octave-mode
# Numerical parameter file (Double_numpar.txt)
# Generated by MTT at Fri Oct 1 12:22:14 BST 1999
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% Version control history
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
# %% $Id$
# %% $Log$
# %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#These are the parameters for the prototype suspension
global rho_1 rho_2 x1 y1 z1 x2 r2 r_1 r_2 y_1 y_2

x1      =      0.100;          % dimension of UpperMass
y1      =      0.041;
z1      =      0.076;
rho1    =      7800;          % density of aluminium
m_1     =      rho1*x1*y1*z1;  % Upper Mass

x2      =      0.1016;        % dimensions of test mass
r2      =      0.0635;
rho2    =      2202;          % density of silica quartz
m_2     =      rho2*pi*r2^2*x2; % Test Mass

g       =      9.80665;       % gravity

r_2     =      62e-6;         % wire radius
r_1     =      89e-6;

l_2     =      0.3700;        % Wire lengths
l_1     =      0.1995;        %

y_2     =      1.65e11;       % Youngs Mod steel 5 thou
y_1     =      1.72e11;

j_1     =      m_1*(z1^2+x1^2)/12; % Moments of inertia
j_2     =      m_2*(r2^2/4+x2^2/12); %

k_2     =      2*y_2*pi*r_2^2/l_2; % Vertical spring constant
k_1     =      2*y_1*pi*r_1^2/l_1; %

vu_1    =      0.003;         % vertical offsets
vl_1    =      0;
vr_1    =      0;
vl_2    =      0.003;
vr_2    =      0.003;

hu_1    =      0;             % horizontal offsets
hl_1    =      -0.0058;
hr_1    =      0.0058;
hl_2    =      -0.0058;
hr_2    =      0.0058;

```

Table 4.13: *Double\_numpar.m: The numerical parameter file for system Double*

## Chapter 5

# The Bond Graph Modelling of Suspensions

### 5.1 Introduction

“Nothing in life is as good or as bad as one first thinks”

Guy de Maupassant

In this chapter various aspects of wire modelling are introduced. It is instructive to introduce a “simple” wire model at which point the deficiencies of this model and possible improvements can be expounded. This will lead to the basic wire framework, as introduced in the previous chapter. Possible extensions to this framework will then be introduced. These extensions allow for bending and/or transverse modes of a wire to be incorporated into specific system models. Moreover this will highlight the need for a modelling technique which enables the interchange of various components without the need for complete remodelling of the overall physical system. Bond graphs fulfill this criterion.

In addition, a method of generating a component’s constitutive relationship directly from a components energy equation is introduced. Since the bond graph methodology is based in the energy domain this is the most appropriate manner by which constitutive relationships

should, where possible, be derived.

## 5.2 An Introduction to Modelling Wires

In this section the restoring force due to wire extension and the fourth order partial differential equation of a stiff beam are introduced. These provide the mathematical foundation for the subsequent hierarchical bond graph models of wires.

### 5.2.1 Restoring Force Due To Wire Extension

A wire under tension provides a restoring force ( $F$ ) given by

$$F = -k \delta l \quad (5.1)$$

where  $k$  is the spring constant and  $\delta l$  is the wire extension. This does assume that the stress, given by

$$\text{Stress} = \frac{\text{Tension}}{\text{Cross Sectional Area Of Wire}}, \quad (5.2)$$

experienced by a wire, is significantly less than the breaking stress, and hence obeys Hooke's law.

The spring constant is derived from the definition of Young's modulus ( $E$ ) given by

$$E = \frac{\text{Stress}}{\text{Strain}} = \frac{F/A}{\delta l/l} \quad (5.3)$$

giving

$$k = \frac{E A}{l} \quad (5.4)$$

where, for a circular wire, the area  $A = \pi r^2$ .



### 5.2.2 Transverse Motion of a Wire Under Tension

A wire under tension can be modelled as a stiff beam under tension ( $T$ ) given by the fourth order partial differential equation (Equation 5.5) [49].

$$T \frac{\partial^2 x}{\partial y^2} - EI \frac{\partial^4 x}{\partial y^4} = \rho S \frac{\partial^2 x}{\partial t^2} \quad (5.5)$$

where,  $E$  is Young's modulus,  $I$  is the cross-sectional moment, (for a circular wire  $I = \pi r^4/4$ ),  $\rho$  is the mass density,  $S$  is the cross-sectional area, and  $x(y, t)$  is the transverse displacement of a point.

Well below the violin mode frequencies a static solution to Equation 5.5 gives a valid description of a wire's bending dynamics. This is because at low frequencies the transverse acceleration of a point approximates to zero, that is

$$\frac{\partial^2 x}{\partial t^2} \approx 0 \quad (5.6)$$

and the terms involving the second and fourth derivatives of  $x$  with respect to  $y$ , dominate, and hence Equation 5.5 reduces to

$$T \frac{\partial^2 x}{\partial y^2} - EI \frac{\partial^4 x}{\partial y^4} = 0 \quad (5.7)$$

However, at violin mode frequencies, Equation 5.5 is dominated by the two second order partial derivatives and hence reduces to the 1-dimensional wave equation (Equation 5.8),

$$T \frac{\partial^2 x}{\partial y^2} = \rho S \frac{\partial^2 x}{\partial t^2} \quad (5.8)$$

Bond graph components which model Equations 5.1, 5.7 and 5.8, to produce restoring forces due to linear extension, bending dynamics and transverse modes respectively, will now be presented.

## 5.3 The Creation of a Basic Wire Framework

Taking advantage of the hierarchical nature of the bond graph methodology facilitates the creation of wire components that can be built in a modular fashion, and hence permits



the easy interchange of variously configured wire models. These wire bond graph models can take many forms, ranging from a light rigid rod where no bending or linear extension is allowed, to a completely flexible wire where bending dynamics, linear extension and transverse modes of vibrations are all modelled.

### 5.3.1 Modelling a Simple Pendulum As a Rigid Rod

The simplest bond graph of a simple pendulum merely consists of a rigid rod (Gawthrop [37]). This bond graph model is equivalent to using a mass element (Section 4.5) with two suspension points, one above and one below the centre of mass. Zero velocity sources are applied to the  $x$  and  $y$  ports of the upper suspension point, indicating that this point is fixed in space and zero force sources are applied to the  $x$  and  $y$  ports of the lower suspension point, indicating that no forces are applied here. Analysis of this system indicates that only one of the mass inertia elements may have integral causality, the other two requiring the derivative form. Choosing integral causality on the inertia element associated with the  $\Theta$  coordinate results in constrained state equations of the polar state space form. However, if either of the other two inertia elements are assigned integral causality a Cartesian state space description results. Yet the linearised transfer functions are identical. This simple pendulum model can, with minor modifications, be used to create a light rod bond graph component.

### 5.3.2 A Simple Light Rod

It is possible to create a simple light rigid rod from a modified two port mass component, as shown in Figure 5.1. Firstly, ports associated with control inputs, the  $x$  and  $y$  **INTF** components, and the restoring force due to gravity are all removed. The named inertia elements  $m_x$  and  $m_y$  are replaced by **SS** elements which impose zero forces, and finally the angular inertia element ( $j$ ) is assigned a very small valued moment of inertia (say  $10^{-6} \text{ Kg m}^2$ ). This inertia element, with integral causality, is retained so as to provide the wire's angular velocity. The velocity can then be integrated, using the remaining **INTF** component

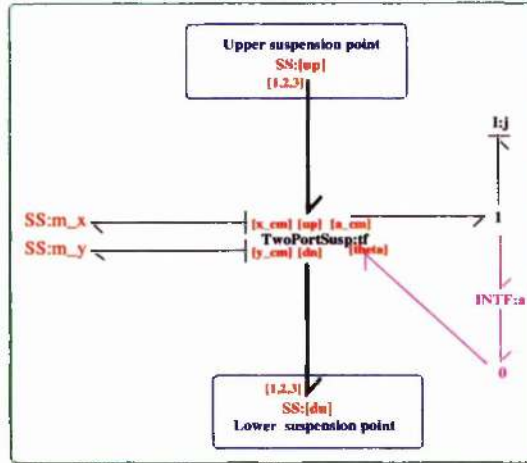


Figure 5.1: The bond graph of a modified two port mass component used to model a light rod

to provide the wire's angle of rotation. Remember, this variable is required by the **EMTF** transformers of the compound component **CoordTrans** (Section 4.5.3). This light rod component may be used in conjunction with various bond graph hinges (Figure 5.2) and masses to produce a more realistic model of multi-stage pendula. Two hinges are shown in Figures 5.2(A) and 5.2(B) respectively, these are used as depicted in Figure 5.2(C). Note, the zero junctions associated with each hinge's  $\Theta$  coordinate have named bonds **SS:[a]** these bonds are attached to the **0** junction with the named **SS** component **SS:[zero]**. This permits the compound component **HingedLightRod** and conjoined components to each have different angles of rotation. The **SpringHinge** component Figure 5.2(B) contains a **C** component which provides a restoring force due to linear extension in the **Y** direction. This restoring force simply implements a restoring force consistent with Hooke's law, ( $F = ky$ ), where  $k$  is the spring constant given by

$$k = \frac{EA}{L} \quad (5.9)$$

where,  $E$  is Young's modulus,  $A$  is the wire's cross-sectional area,  $L$  the wire's length and  $y$  the extension from the equilibrium position in the **Y** direction.

Using the **SpringLightRod** component the "simple" single stage pendulum of Figure 5.3 can be produced. The advantage of this model is that further degrees of freedom are intro-

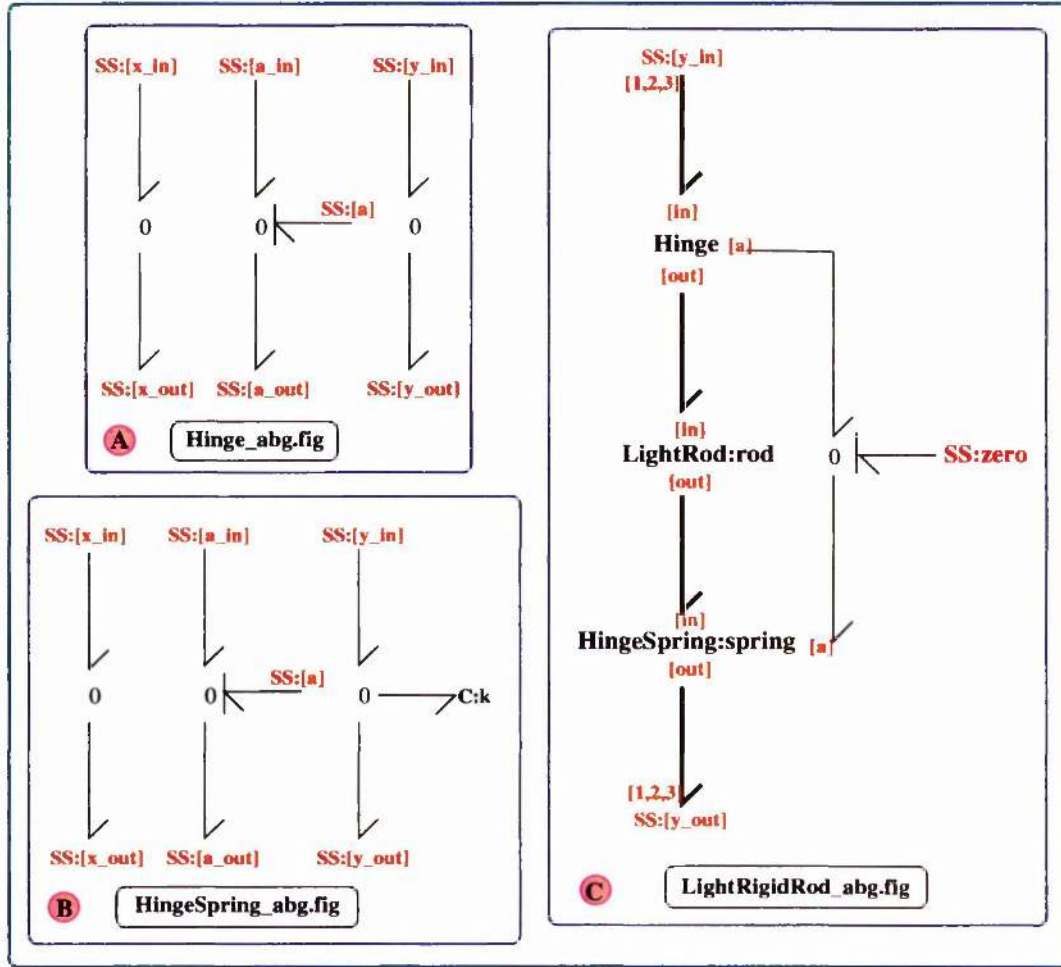


Figure 5.2: **Hinge**, **HingeSpring** and **HingedLightRod** bond graph components. **HingedLightRod** utilises **LightRod** (Figure 5.1) and the two types of hinge component to produce a “simple wire” component.

duced: One extra degree of freedom through the incorporation of the vertical spring within component **HingeSpring**, and a further degree of freedom from the independence of mass and wire angular rotations.

At this point it is worth noting that the development of this “simple” wire component is rather more complicated than one might have expected. Already six components are utilised and yet wire dynamics due to bending and transverse modes of vibrations have still to be modelled. The complexity arises through the need to determine the angular rotation and

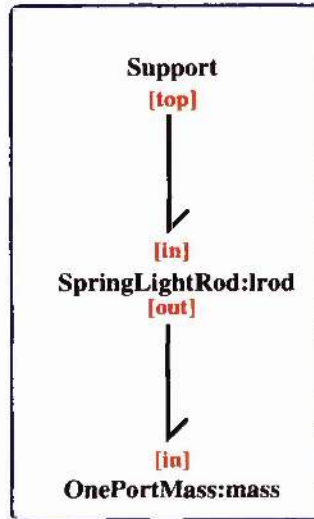


Figure 5.3: A bond graph model of a simple pendulum with two degrees of freedom. Key components are *SpringLightRod* and a single port mass element.

linear extension of a wire. A more elegant method for the calculation of these variables was presented in Section 4.6 (Page 69). This forms the basic framework for all subsequent wire models. The advantage of this model is that only one component is required to calculate a suspension's angular rotation and linear extension. Further, implicit in this formalism, is that a wire no longer has a fixed length. In addition, as will be shown, it is easy to incorporate appropriate wire dynamics into this framework. The next sections detail how both wire bending and transverse modes of vibration are modelled.

## 5.4 Introduction to Modelling the Bending Dynamics of a Wire

The basic framework of a wire bond graph, as introduced in Section 4.6 (Page 69), is shown in Figure 5.4. Embedded within this component is a subcomponent **LinExt**. This subcomponent models the restoring force due to the wire's linear extension (Section 4.6.1 – Page 69). To include the bending dynamics the subcomponent **LinExt** is replaced by a more sophisticated model, as follows.



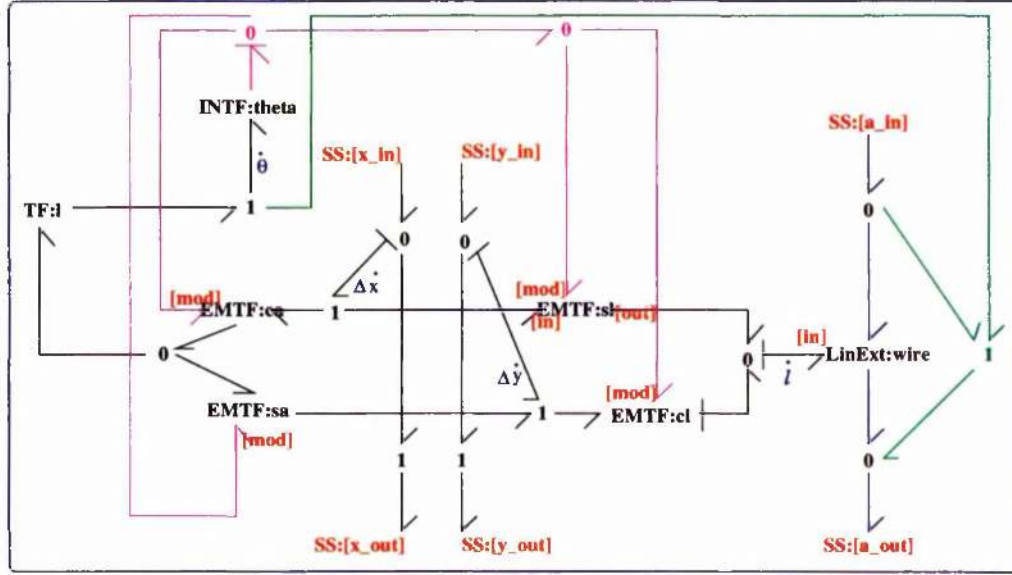


Figure 5.4: The basic framework for the modelling of wire dynamics.

#### 5.4.1 Bending Energy of a Flexible Wire

The fourth order ordinary differential equation of a tensioned beam (Equation 5.5) was introduced in Section 5.2 and is re-stated in Equation 5.10.

$$T \frac{\partial^2 x}{\partial y^2} - EI \frac{\partial^4 x}{\partial y^4} = \rho S \frac{\partial^2 x}{\partial t^2} \quad (5.10)$$

where,  $T$  is the wire tension,  $E$  is Young's modulus,  $I$  is the cross-sectional moment, (for a circular wire  $I = \pi r^4/4$ ),  $\rho$  is the mass density,  $S$  is the cross-sectional area,  $x(y, t)$  is the transverse displacement of a point along the wire.

Remember, at frequencies well below the violin modes a static solution to Equation 5.10, provides a valid description of the bending dynamics, and hence Equation 5.10 reduces to

$$T \frac{\partial^2 x}{\partial y^2} - EI \frac{\partial^4 x}{\partial y^4} = 0 \quad (5.11)$$

A general solution to Equation 5.11 is given by

$$x(y) = A + By + Ce^{ky} + De^{-ky} \quad (5.12)$$

where the constants  $A, B, C$  and  $D$  are determined by the boundary conditions and

$$k = \sqrt{T/(YI)} \quad (5.13)$$

where,  $\lambda = 1/k$  is the characteristic bending length and determines the arc length over which the bending occurs. The general boundary conditions are:-

$$\begin{aligned} x(0) &= x_1 \\ x(l) &= x_2 \\ \frac{dx(0)}{dy} &= \theta \\ \frac{dx(l)}{dy} &= \phi \end{aligned} \quad (5.14)$$

where,  $x_1, x_2, \theta, \phi$  are defined in Figure 5.5.

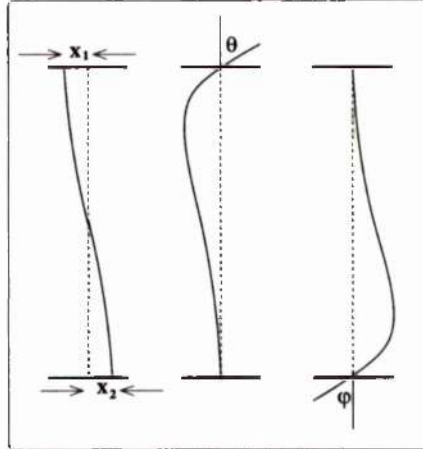


Figure 5.5: General boundary conditions for bending of a stiff beam under tension

The elastic energy stored in a wire is given by Equation 5.15 [36]

$$E = \frac{1}{2} \int_0^l YI \left( \frac{\partial^2 x}{\partial y^2} \right)^2 dy \quad (5.15)$$

Hence by substituting the boundary conditions into Equation 5.12 the four constants ( $A, B, C$  and  $D$ ) can be found. Equation 5.12 can then be substituted into the energy equation, thus giving the bending energy stored in a tensioned wire as a function of the variables,

$x_1, x_2, \theta, \phi$ . To solve this equation analytically, without the aid of a computer, is almost impossible without making many approximations. However, by using an algebraic software package, e.g. REDUCE<sup>1</sup>, a complete analytical solution can be made. Whilst performing appropriate transformation MTT utilises Reduce and hence the bond graph component modelling bending dynamics can use the full analytical solution rather than some approximation.

From Equation 5.15 the restoring force, as a function of each variable, is given by Equation 5.16.

$$F_i = -\frac{\partial E}{\partial i} \quad (5.16)$$

where,  $x_1, x_2, \theta, \phi$  is to be substituted for the index  $i$ .

This is of the form<sup>2</sup>,

$$F_i = \psi(x_1, x_2, \theta, \phi) \quad (5.17)$$

and is ideally suited to the creation of a multiport bond graph flow store (C component). This bond graph component will now be presented.

#### 5.4.2 The Bond Graph Model of a Bending Wire

As stated earlier (in Section 3.4.1) a simple spring can be represented in the bond graph domain by a capacitive C component. Remember the constituent relationship for a C component is given by Equation 5.18

$$e = \Phi_C(q) \quad \text{where} \quad q = \int_{f_0}^f f dt \quad (5.18)$$

and in the linear case

$$e = \frac{q}{C} \quad (5.19)$$

---

<sup>1</sup>Reduce is a commercially available symbolic algebra package which is a necessary component of MTT. It is used to construct constituent relationships and to carry out symbolic manipulation during MTT routines (for example in the generation of differential algebraic equations)

<sup>2</sup>It should be noted that this equation constitutes a linear relationship. However, the method is generic and as such the bond graph methodology is able to cope with nonlinear relationships, see Section 3.9



where  $C$  is known as the *capacitance*.

Notice that Equation 5.18 is of the same form as Equation 5.17. Therefore a four port bond graph  $C$  component, one port for each of the variables  $x_1, x_2, \theta, \phi$ , in conjunction with the constitutive relationship<sup>3</sup> of Equation 5.16, form the basis of a bond graph component which models the dynamics of a wire due to bending. Figure 5.6 shows the four port  $C$  component where each bond has an associated flow and effort variable.  $\dot{x}_1, \dot{x}_2, \dot{\theta}, \dot{\phi}$  are the time derivatives of the generalised displacements, i.e., the flow variables, and  $f_1, f_2, \tau_1, \tau_2$  are the respective effort variables (corresponding to forces ( $f$ ) or torques ( $\tau$ )).

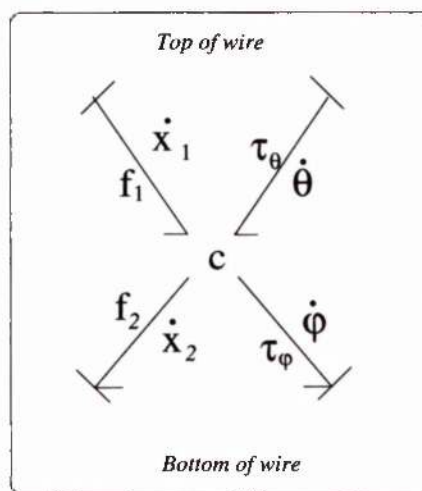


Figure 5.6: A four port bond graph  $C$  component

However, this four port bond graph can be reduced to a more concise bond graph, which also contains the restoring force due to linear extension. This is possible because the bending component will be embedded within the bond graph of Figure 5.4 and as such there is a change in boundary conditions. The new boundary conditions set  $x_1 = x_2 = 0$ ; in effect the bond graph which constitutes the basic wire framework (Figure 5.4) performs a coordinate transformation. The new boundary conditions are shown in Figure 5.7 and therefore the four port bond graph of Figure 5.6 can be reduced to a three port component, as shown in Figure 5.8, two ports associated with bending and the other with the restoring force due

<sup>3</sup>Within MTT a component's constitutive relationship is defined in an MTT file ComponentName.cr. MTT provides the ComponentName.cr files for each of the components found within an MTT distribution. However, it is possible for an individual user to produce constitutive relationship files of their own. These files are written using the Reduce syntax.

to wire extension.

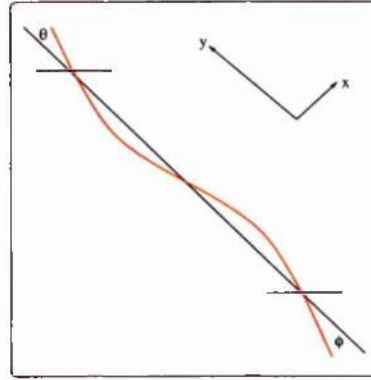


Figure 5.7: *Effective boundary conditions for an embedded **Bending** component*

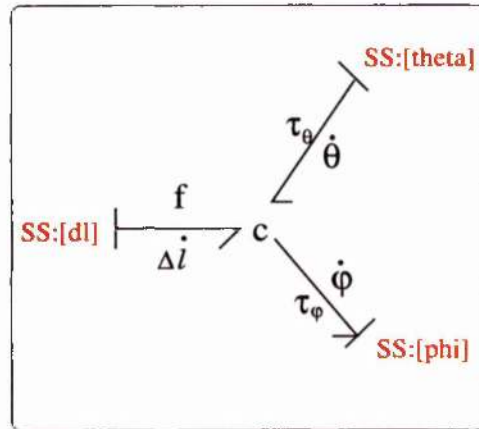


Figure 5.8: *A 3 port bond graph component combining extension and bending characteristics of a flexible wire*

This component has been given the name **Bending\_abg.fig** and is embedded within the component **Wire\_abg.fig** (Figure 5.9). Note, the only difference between this and the earlier **Wire\_abg.fig** is that the subcomponent **LinExt:wire** has been replaced by **Bending:wire** (c.f. Figures 5.4, Page 91 and 5.9, Page 96). That is the introduction of the bending dynamics is achieved by simply interchanging the component **LinExt** with component **Bending** without the need for remodelling the complete system. MTT can then be invoked to create alternative model representations (for example state space matrices). Moreover, with bond graphs based in the energy domain, the generation of the constitutive relationship di-

rectly from the energy equation is the most appropriate method for the derivation of such relationships. This bending model is valid for frequencies well below the transverse mode frequencies. The modelling of a wire's transverse modes will now be detailed.

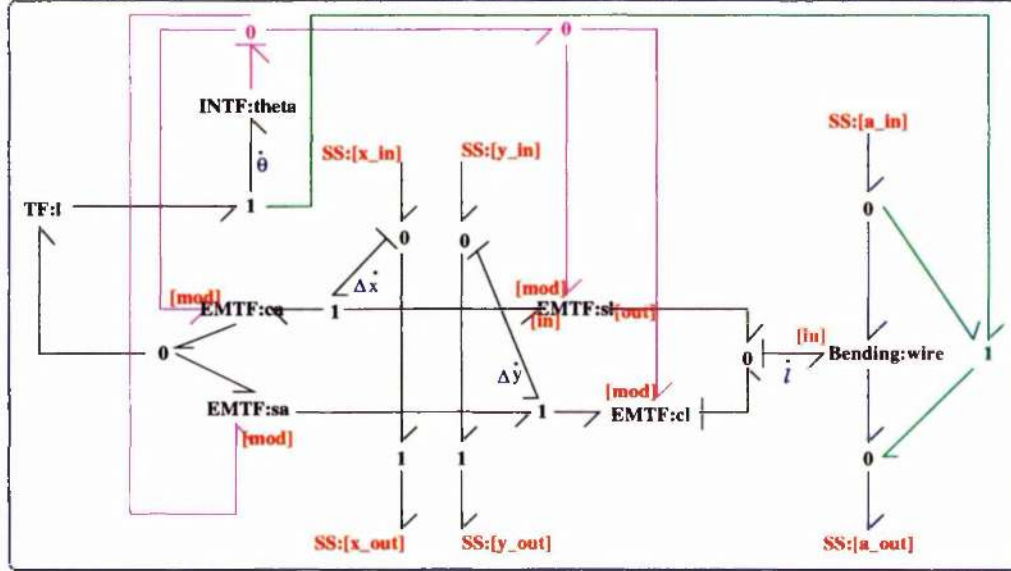


Figure 5.9: The inclusion of bending dynamics into the basic wire model. Note that the only difference between this and the component *LinExtWire\_abg.fig* is that *LinExt:wire* has been replaced by *Bending:wire*

## 5.5 Modelling Transverse Modes of a Tensioned Wire

At first sight there would appear to be two approaches to modelling the transverse modes of a wire: A lumped approach and a finite modal method. However, it will be shown that the finite modal method, although appropriate in other circumstance, is inappropriate for modelling wires in the context of pendulum suspensions. Hence a lumped representation, although not ideal, is presented and ultimately used for modelling violin modes. Models will be compared with experimental results.

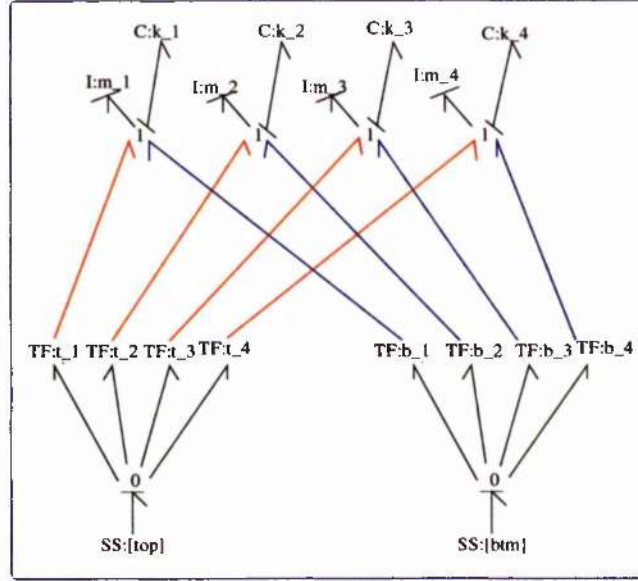


Figure 5.10: The bond graph of the modal method of modelling the modes of a continuous system. Here four modes are to be model with two force inputs.

### 5.5.1 Finite Modal Method of Modelling Transverse Modes

The bond graph finite modal method of modelling transverse vibrations for continuous systems was presented by Margolis and is comprehensively covered in both book and paper formats [44] [50]. A typical modal bond graph can be seen in Figure 5.10. This is based upon Equation 5.20

$$T \frac{\partial^2 x}{\partial y^2} = \rho S \frac{\partial^2 x}{\partial t^2} \quad (5.20)$$

Where  $T$  is the wire tension,  $E$  is Young's modulus,  $I$  is the cross-sectional moment, (for a circular wire  $I = \pi r^4/4$ ),  $\rho$  is the mass density,  $S$  is the cross-sectional area,  $x(y, t)$  is the transverse displacement of a point along the wire. A separation of variables is then performed by assuming that,

$$x(y, t) = X(y)f(t) \quad (5.21)$$

giving two total differential equations, one spatial the other temporal. Then, for given boundary conditions, mode shapes ( $X_n(y)$ ) and frequencies ( $\omega_n$ ) can be determined, where



the subscripts  $n$  denote the  $n^{\text{th}}$  mode. The mode shapes are orthogonal, in that

$$\int_0^L X_n(y)X_m(y) dy = 0 \quad (5.22)$$

This property is used to derive the modal mass ( $m_m$ ) given by,

$$m_m = \int_0^L \rho A X_m^2(y) dy, \quad m = 1, 2, \dots \quad (5.23)$$

where  $\rho$  is the mass density and  $A$  is the cross-sectional area, where for a circular wire

$$m_m = \frac{\rho(\pi r^2)L}{2} \quad (5.24)$$

where  $L$  is the length of the wire and  $r$  its length. The modal stiffness ( $k_m$ ) is given by,

$$k_n = m_n \omega_m^2 \quad (5.25)$$

These are precisely the parameters,  $m_n$  and  $k_n$ , of components **I** and **C** respectively, used in the modal bond graph of Figure 5.10. The parameters of the **TF** components are simply the values of the mode shapes  $X(y)$  evaluated at the point at which a force is applied. This analysis assumes that system inputs are forces. Margolis then proceeds to investigate the case where system inputs are velocities.

A consequence of having velocity inputs is that either inertia elements are forced to have differential causality or, to avoid this form of causality, extra modal compliances are added, one for each velocity input. Margolis advocates the use of extra compliances to bypass the extra computational effort arising from derivative causality. However, by employing a computer to carry out necessary transformations derivative causality, in general, is not a problem.

However, within the context of modelling suspension wires this finite modal component has two velocity inputs imposed upon it, and as a consequence algebraic loops need to be solved [50]. MTT provides a mechanism for solving simple algebraic loops [51]. However, the algebraic loops, associated with this two velocity input modal component, involve the derivatives of inputs and states: Finding solutions to these is not a trivial matter. Moreover, since the use of this component requires prior knowledge of the boundary conditions, and input/output configuration, it was deemed unsuited to the modelling of a suspension wire's violin modes. As such, an alternative approach to modelling violin modes will now be presented.

### 5.5.2 A Lumped Method of Modelling Transverse Modes

This is the preferred method of implementing transverse vibrations of a tensioned wire. This is because, within the context of modelling suspensions wires, the lumped method provides a significant increase in the accuracy of generated mode frequencies vis-a-vis the modal method. Furthermore, this component can be incorporated into a system model without any prior knowledge of the overall system configuration (causality etc).

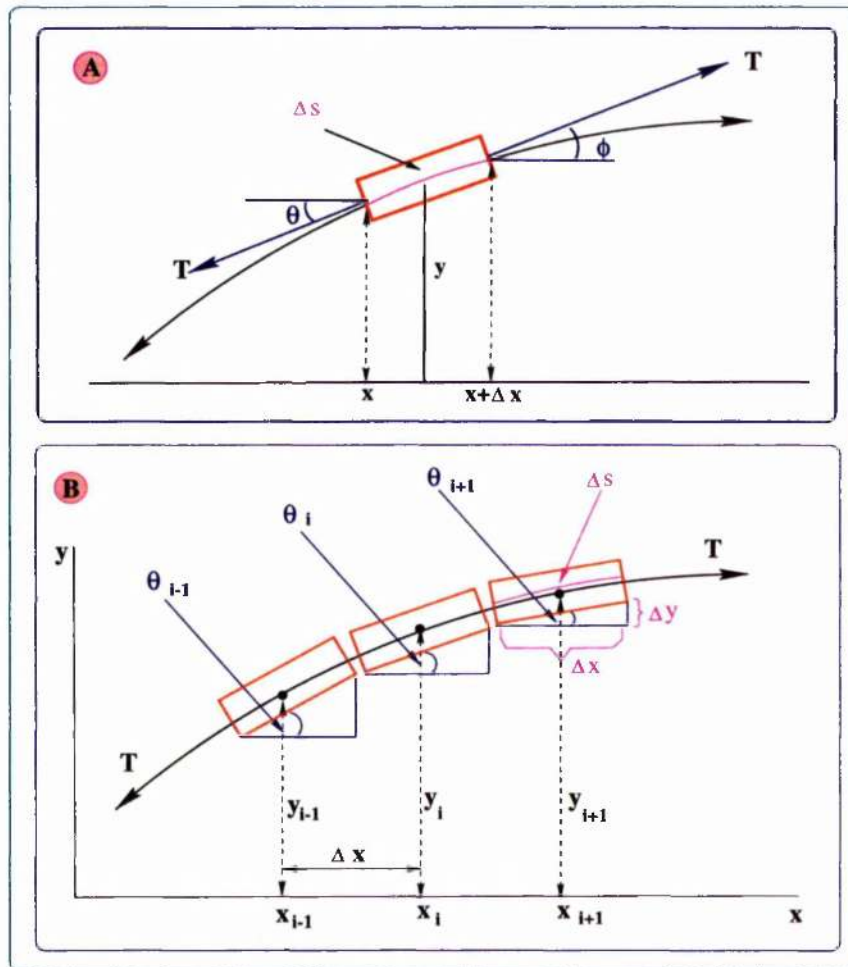


Figure 5.11: Force diagram for a short segmented massive wire undergoing transverse vibrations

### 5.5.3 Mathematical foundation

In Figure 5.11(A) a single segment of a wire under tension is shown. It is assumed that the displacement of any wire segment, with respect to its equilibrium position, is small. The two angles  $\theta$  and  $\phi$  are the direction tangents at  $x$  and  $x + \Delta x$  respectively. The net forces acting on this segment are given by

$$\begin{aligned} F_y &= T \sin(\phi) - T \sin(\theta) = \rho A \Delta x \frac{\partial^2 y}{\partial t^2} \\ F_x &= T \cos(\phi) - T \cos(\theta) \end{aligned} \quad (5.26)$$

where  $\rho$  is the linear mass density and  $A$  is the cross-sectional area. By making the assumption that the angular rotation of any segment is small, such that the arc length of a small segment  $\Delta s \approx \Delta x$ , and therefore  $\sin(\theta) \approx \tan(\theta)$  Equation 5.26 then becomes

$$\begin{aligned} F_y &\approx T \left[ \frac{\partial y}{\partial x} \Big|_{x+\Delta x} - \frac{\partial y}{\partial x} \Big|_x \right] = \rho A \Delta x \frac{\partial^2 y}{\partial t^2} \\ F_x &\approx 0 \end{aligned} \quad (5.27)$$

where the subscripts refer to the points at which the gradients are evaluated, giving the 1-dimensional wave equation [52],

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2} \quad (5.28)$$

where

$$c = \sqrt{T/\rho} \quad (5.29)$$

is the phase velocity of the wave.

### 5.5.4 Bond Graph Synthesis

Referring to Figure 5.11(B), the creation of a bond graph lumped approximation of the 1-dimensional wave equation follows. First let the transverse component of force due to the wire tension  $T \sin(\psi) = F$ . Then from Equation 5.26

$$F_{i+1} - F_i = \frac{d}{dt} p_i \quad (5.30)$$



where  $p_i$  is the momentum of the  $i$ th lump

$$p_i = \rho A \Delta x \dot{y} \quad (5.31)$$

and from Equation 5.27

$$F_i = \frac{T}{\Delta x} q_i \quad (5.32)$$

where  $q_i$  is the relative displacement between the  $i$ th and  $(i - 1)$ th lump, i.e.,

$$q_i = y_i - y_{i-1} \quad (5.33)$$

Hence for the internal elements of the lumped representation the following equations apply

$$\frac{d}{dt} p_i = \frac{T}{\Delta x} (q_{i+1} - q_i) \quad (5.34)$$

$$\frac{d}{dt} q_i = \frac{p_i - p_{i-1}}{\rho A \Delta x} \quad (5.35)$$

The equations for the first and last elements depend on the input causalities to the overall lumped representation. With reference to Figure 5.12 the components **SS:[one]** and **SS:[n]** provide inputs and outputs at the first and  $n^{\text{th}}$  elements respectively. With velocity inputs the first component has the following equation

$$\frac{d}{dt} q_1 = V_{one} - \frac{p_1}{\rho A \delta x} \quad (5.36)$$

Meanwhile, the  $n$ th component will have differential causality and hence does not contribute a state to the overall system model. The force output at **SS:[n]** is then given by

$$F_n = \frac{T}{\Delta x} q_n - \rho A \Delta x \dot{V}_n \quad (5.37)$$

With force inputs the first element now has differential causality and hence the velocity output at **SS:[one]** is

$$V_{one} = \frac{1}{\rho A \Delta x} p_i + \frac{\delta x}{T} \dot{F}_{one} \quad (5.38)$$

and the  $n$ th component has the following equation

$$\frac{d}{dt} p_n = \frac{T}{\Delta x} q_n - F_n \quad (5.39)$$



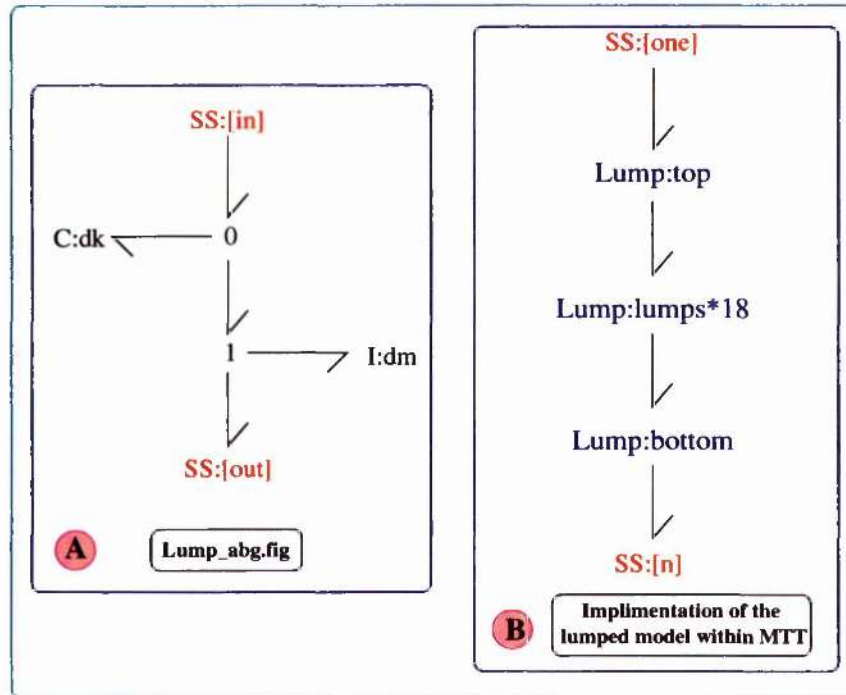


Figure 5.13: *The implementation of the lumped model within MTT*

appropriate amount. This is achieved by replacing “m” by an appropriate positive integer within the bond graph of `Lumped_abg.fig`.

In Table 5.1 the `lbl` file for the lumped representation of transverse modes is detailed. Note that although the component **Lump:lump\*m** appears in the acausal bond graph the multiplicity (`*m`) should not be included in the declaration of the component within the label file.

Now the drawback with this lumped representation is that by increasing the number of lumps, to increase the accuracy of the lower frequency modes, introduces an increasing number of inaccurate higher order modes. The number of lumps that should be used is a decision for the modeller.

```

%% Label file for system Lumped (Lumped_lbl.txt)
%SUMMARY Lumped

%DESCRIPTION In the Figure of Lumped there is some text
%DESCRIPTION Lump:lumps*m where m is the number of lumps
%DESCRIPTION you require in your model

% *****
% %% Version control history
% *****
% %% $Id$
% %% $Log$
% *****

% Port aliases
%ALIAS in in
%ALIAS out out

% Argument aliases

%ALIAS $1 dk
%ALIAS $1 dm

%% Each line should be of one of the following forms:
% a comment (ie starting with %)
% component-name cr_name arg1,arg2,...argn
% blank
% ---- Component labels ----
% lump none dk;dm
% lumpB none dk;dm
% LumpT none dk;dm

% Component type SS
% [in] SS external,external
% [out] SS external,external

```

Table 5.1: The label file for component *Lumped*



### 5.5.5 Comparison of Lumped model with physical system

In Table 5.2 the mode frequencies, as generated by the lumped models with 2, 4 and 16 lumps, are compared with the solution to the 1-dimensional wave equation (with fixed end boundary conditions) and with experimental results. It should be noted that there is some variation in the measured fundamental modes of each physical wire. This is because sets of wires, between successive masses, are subject to differing boundary conditions. Implicit assumptions are that each group of wires has precisely the same length, and the tension in each wire is exactly the same. It is further assumed that each wire is clamped to a pendulum mass such that the wire breaks away from its clamp normal to the clamps horizontal surface. Notice with only 2 lumps the generated fundamental mode is  $\approx 90\%$  of the value generated

Modes	Experiment	Wave Eqn	2 lump	4 lump	16 lump
Upper Wires	504.9 506.4	505.61	455.21	492.71	504.79
Lower Wires	193.1 205 207.6 219.9	204.56	184.17	199.35	204.23

Table 5.2: *Comparison of mode frequencies: Experiment Vs 1-d Wave Equation Vs Lumped Method*

by the 1-dimensional wave equation. However the difference falls to  $< 3\%$  with a 4 lumped representation. Within the context of modelling suspensions 4 modes would suffice as any error within the bond graph model is outweighed by the spread in the measured values.

This lumped representation of transverse modes is similar to that used by Margolis in the analysis of a longitudinally vibrating bar in the sense that it also reproduces the 1-dimensional wave equation. Hence, should a model of longitudinal vibrations be required then the above model can be employed with the constant of proportionality for the stiffness elements changed from  $T/\Delta x$  to  $EA/\Delta x$ , where  $E$  is Young's modulus and  $A$  the cross-sectional area. The mass elements have the same constants of proportionality in both models. (See Margolis [44]).

Now, since a wire is rigidly clamped to a mass, the wire, at its attachment point, must have the same velocity as the mass. Hence, to embed this component within the general wire

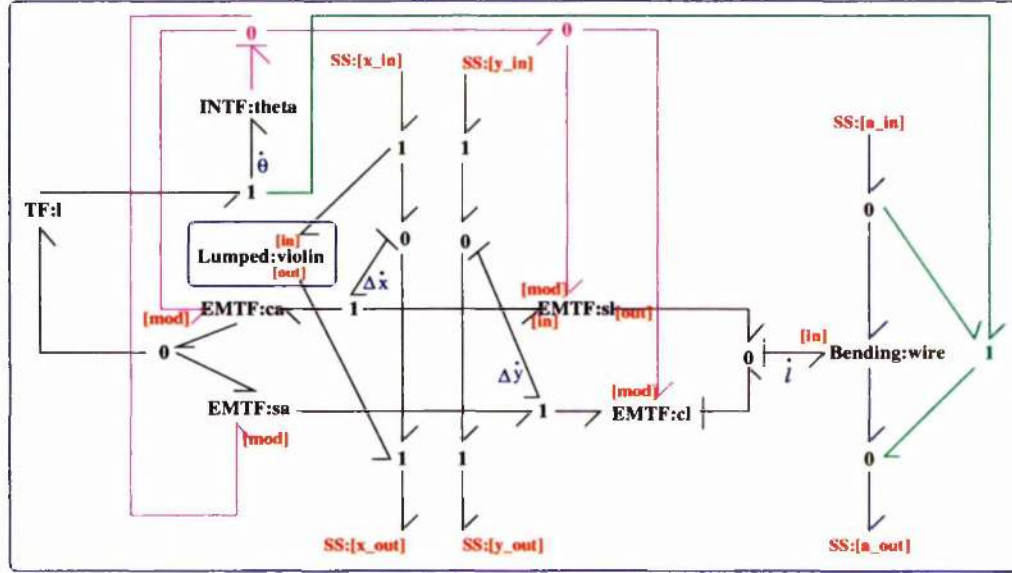


Figure 5.14: Hierarchical bond graph of the component **Wire** incorporating the **Lumped** representation of transverse modes

framework its two ports should be connected to individual **1** junctions of the X coordinate, as shown in Figure 5.14. Further, the modeller still has the choice as to whether the component **LinExt** or **Bending** be used in conjunction with this lumped representation.

## 5.6 Conclusions

In this chapter the hierarchical bond graph models for the modelling of suspension wires were introduced. These components permit the inclusion of:

- restoring forces due to linear extension of a wire,
- wire bending dynamics and,
- the transverse modes of vibration

Furthermore, two possible models for the modelling of transverse modes of a wire were produced and evaluated. It was found that the most appropriate method for modelling these

modes, within the context of suspension wires, was the lumped representation and as such it is recommended that this bond graph component be used. This is because its use does not require any prior knowledge of system causality. Also, it should be appreciated that the bond graph of the lumped wire representation is founded upon the same mathematical analysis as the standard PDE. In addition, since the **Wire** model is constructed in a modular fashion the full advantages of hierarchical bond graphs can be exploited. In subsequent chapters these wire models will be incorporated into pendulum models, which in turn are used in the design of local and global controllers (Chapter 6 and Chapter 7 respectively).



## Chapter 6

# Local Control

### 6.1 Introduction

As stated in Chapter 1, the main optics of the gravitational wave detector are suspended from multi-stage pendula. This affords isolation from seismic vibrations. However, a consequence of using pendula is that seismic motion at the pendulum resonant frequencies is resonantly enhanced. To counteract this phenomenon the pendula are actively controlled. Since this control scheme acts independently, on individual multi stage pendula, it is known as local control. The current control law is an analogue implementation designed by S. Killbourn [48].

In this chapter a digital implementation of the current analogue control law is introduced. This is used to assess the feasibility of digital controller implementation. The analogue and digital implementations are compared and having shown that a digital implementation can satisfactorily replicate the analogue performance an alternative method for designing controllers will be detailed and evaluated. Here a model based observer controller will be designed in the bond graph domain.

Before controller designs are discussed the physical system will be introduced. This will include sections on sensor and actuator design.

## 6.2 Experimental Setup

In Figure 6.1 a photograph of the double pendulum system is shown. This photograph shows the position of the collocated sensors and actuators. For clarity a schematic of this two stage pendulum is shown in Figure 6.2. Points to note are:

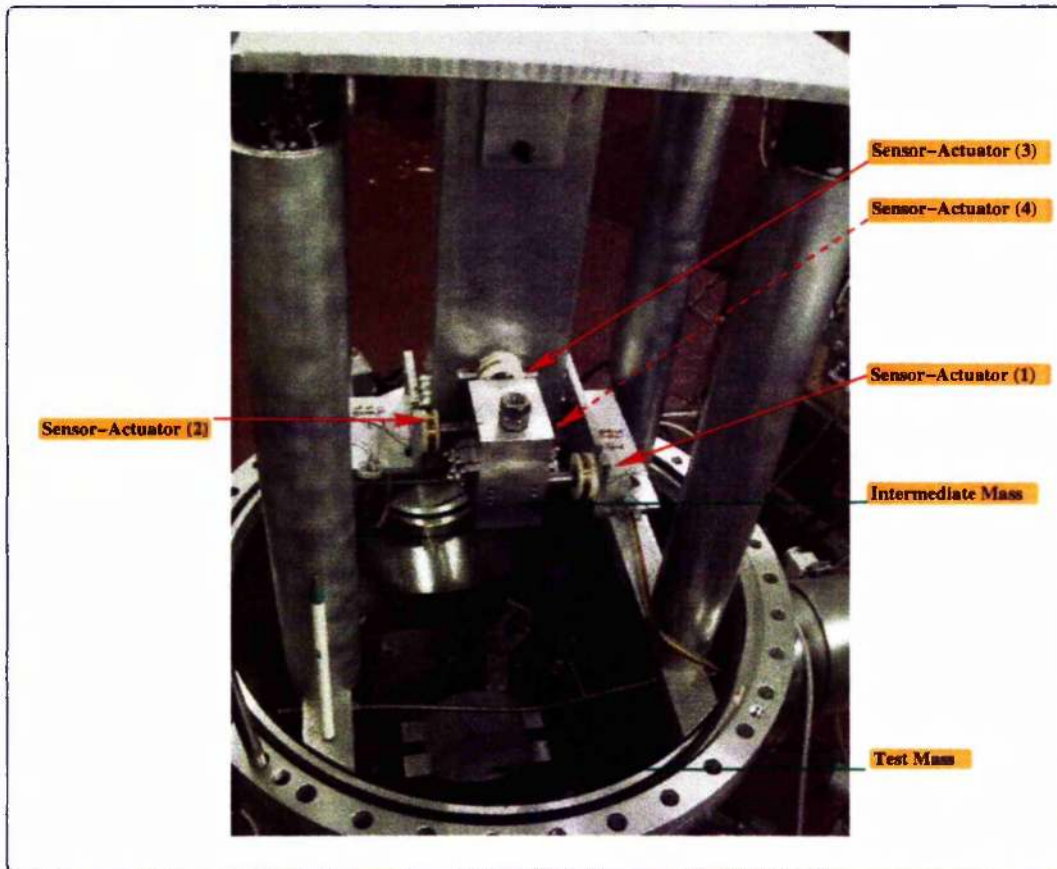


Figure 6.1: A photograph of the double pendulum system

- The two stage pendulum consists of a fused silica test mass ( $0.865\text{ kg}$ ) and an aluminium intermediate mass ( $0.8025\text{ kg}$ ).
- The intermediate mass is suspended by two wires (length  $0.375\text{ m}$ , radius  $62\text{ microns}$ ).
- The test mass is suspended by two loops of wire (length  $0.298\text{ m}$ , radius  $62\text{ microns}$ ).

## SCHEMATIC OF DOUBLE PENDULUM

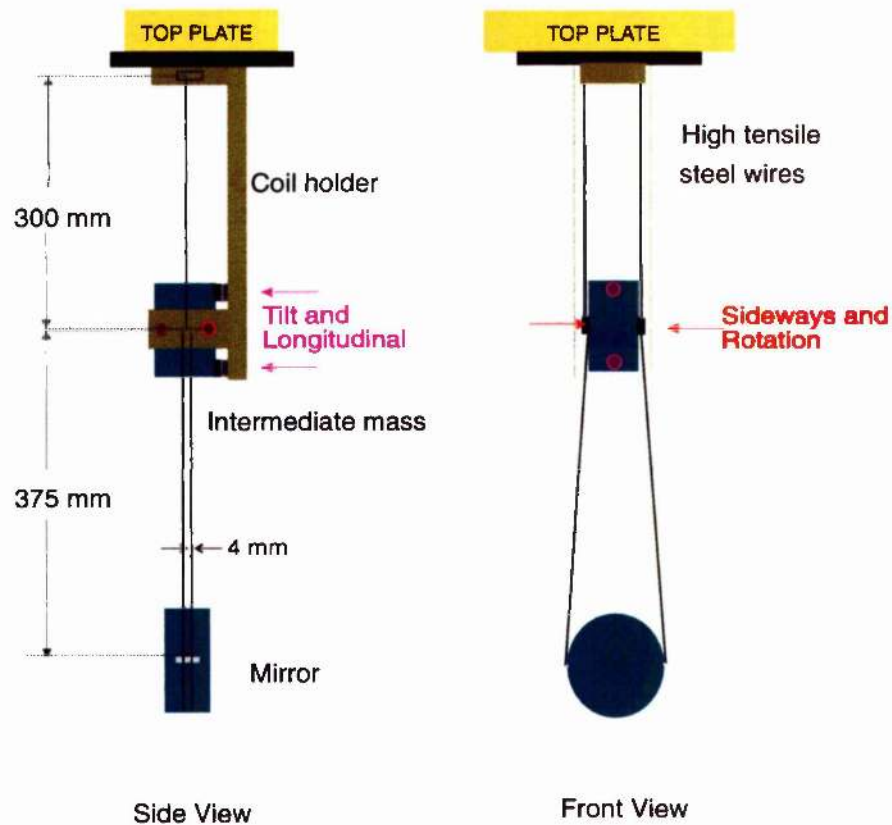


Figure 6.2: Schematic of physical double pendulum

- As can be seen in Figure 6.2, two sets of sensor/actuator pairs are used in the damping of sideways and rotational modes, and another two are used for the damping of longitudinal and tilt modes.

### 6.2.1 Sensors and Actuators

A schematic of the sensor actuator pairs can be seen in Figure 6.3.

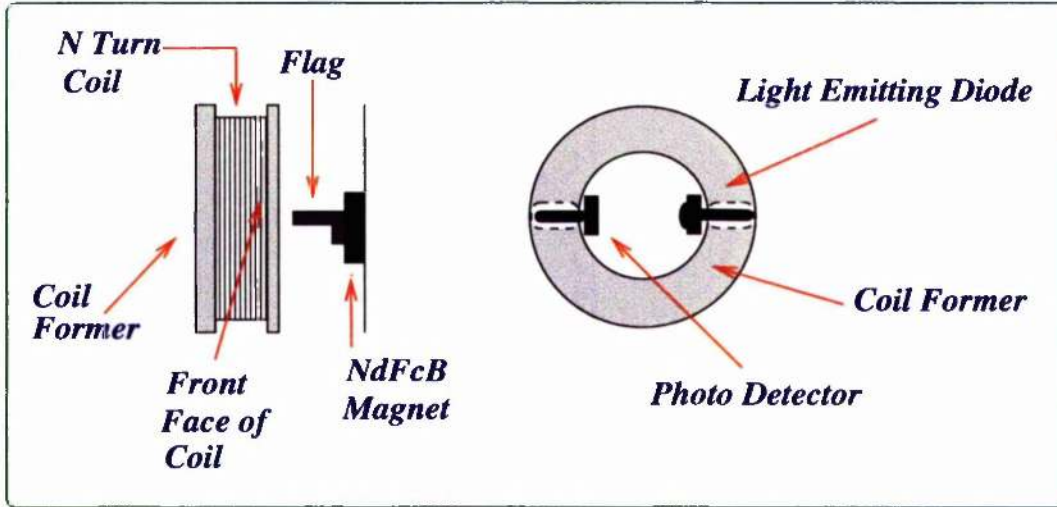


Figure 6.3: A schematic of the sensor actuator combination

### Sensors

The sensors measure mass displacement and consist of an LED (SFH487) illuminating a photodiode (BPW345). An opaque flag, attached to the mass, partially obscures the light beam and hence variations in mass displacement create changes in photocurrent. The photosensitive region extends over 3.5 mm. Each LED provides a flat response across an arc of  $30^\circ$ , with an error of  $+0\%$  and  $-10\%$ , centred about the tip of the led. Therefore, for a photodiode–LED separations greater than 6.53 mm the non uniform response of the sensor is reduced to a minimum.

### Actuators

Feedback forces are provided by a coil/magnet combination. The magnet is attached to the mass and a force, generated by passing a current through the coil, acts on the magnet. Obviously, for a given current through a coil, the force experienced by the magnet depends on their relative positions. The force along the axis of the coil is given by

$$F_x = p \frac{dB_x}{dx} \quad (6.1)$$



where  $p$  is the magnetic dipole and  $B_x$  is the axial magnetic field produced by the coil. For a coil of  $N$  turns, and radius  $a$ , the magnetic field is given by

$$B_x = \frac{\mu I N a^2}{2(a^2 + x^2)^{3/2}} \quad (6.2)$$

where  $\mu$  is the permeability of space and  $I$  is the current through the coil. The variation of force for a given coil magnet separation is shown in Figure 6.4. The origin of this graph corresponds to the front face of the magnet and the plane situated at the centre of the coil and perpendicular to the axis of the coil. Notice that if the coil-magnet separation is  $\sim 6$  mm then across the effective range of the sensor the variation in the applied force, for a given current, is  $\approx 10\%$

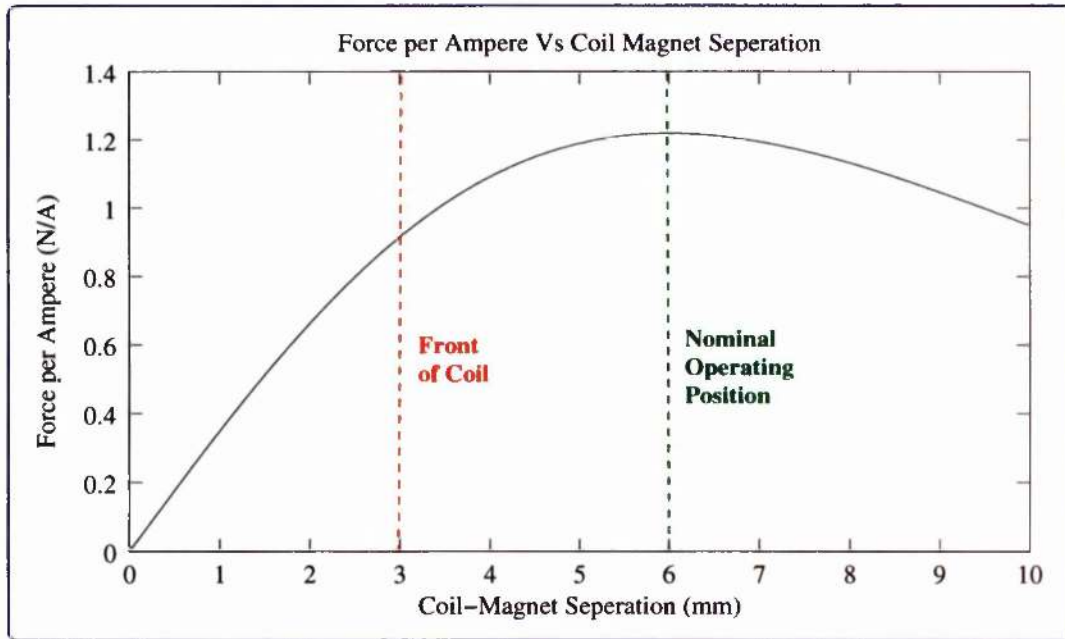


Figure 6.4: Variation of force with respect to coil magnet separation

In general the motion of the intermediate mass is expected to be tens of microns and hence, about the operating point, the response of the sensor/actuator combination is expected to be flat.

### 6.3 The Existing Analogue Local Controller

The current analogue controller is detailed in S Kilbourn's thesis [48] and basically consists of differentiation to 12 Hz. A transitional filter is used to produce a flat response between 0.7 Hz and 2 Hz which reduces the gain ratio between the upper and lower modes. However this results in the phase at the lower mode not being at its optimal phase of  $+90^\circ$ . To restore this phase, extra differentiation below the lower resonance is introduced. That is from 0.1 Hz to 0.7 Hz. This constitutes the damping component of the controller. However to reduce the effects of high frequency sensor noise, two resonant filters, one with a nominal resonant frequency at 15 Hz and Q factor of 3, the other at 18 Hz and Q factor of 4, and a single-pole low-pass filter at 12 Hz, are added. The bode plot of this controller can be seen in Figure 6.5 and is implemented using an analogue circuit.

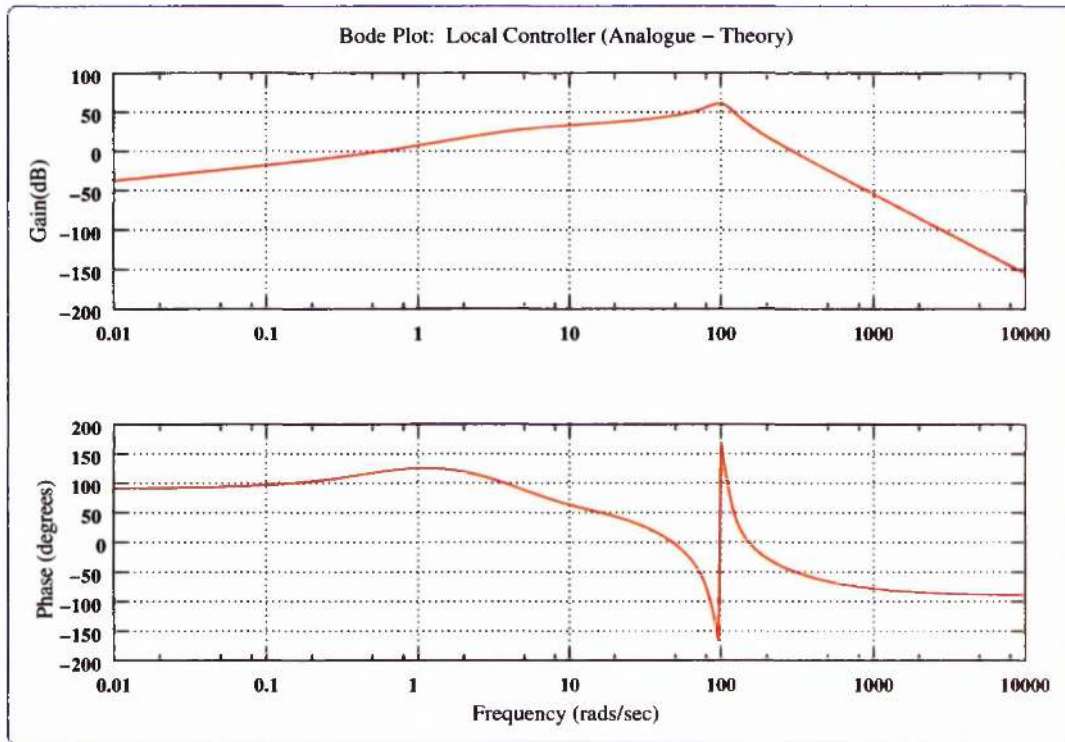


Figure 6.5: Theoretical bode plot of analogue local controller



## 6.4 Digital Implementation of the Existing Analogue Control

In this section, the digital implementation of the existing analogue local controller is detailed. First the basics of digital implementation are introduced, this includes sections on digital controller design methodologies and forms of computer implementation. Finally, the analogue and digital controllers are compared.

### 6.4.1 Introduction

A block diagram of a basic sampled data system can be seen in Figure 6.6, where  $G(s)$  is a continuous representation of the plant to be controlled,  $D(z)$  represents a discrete transfer function of a digital compensator and the blocks marked ADC and DAC are the analogue-to-digital and digital-to-analogue converters respectively. Here the continuous error signal is sampled by the ADC, at a given sample rate, to produce a digital error signal  $e(z)$  (Note: It is also common practice to sample the output  $y(s)$  and the reference signal  $r(s)$ , and then to sum these directly to produce a digital error signal  $e(z)$ ). A digital output  $u(z)$  is generated, according to the algorithm defined by the discrete transfer function  $D(z)$ , which is converted into a continuous output  $u(s)$  using the DAC. This continuous output is, in general, achieved using a zero order hold (ZOH). Here the DAC voltage is held constant at the latest digital value until replaced, and held constant, at the next sample instant.

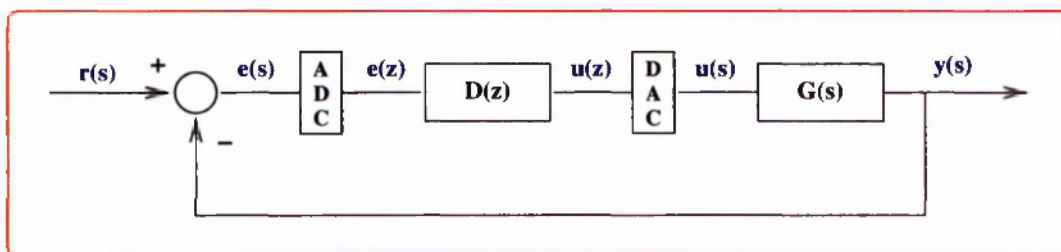


Figure 6.6: A block diagram representation of a sampled data system

### 6.4.2 Digital Design in the Continuous Domain

There are two approaches to the design of digital controllers, either design in the continuous domain, and then convert to the digital domain, or design solely in the digital domain. Thus far, since control design in the continuous domain is mature and well understood, the former design process has been the most popular. Moreover, with the increasing speed of processors, providing higher sampling rates, digital implementations can more accurately replicate a continuous design. In this thesis, the design in the continuous domain and subsequent conversion to a digital representation is undertaken. This is a valid approach because the bandwidth of a local controller is approximately 20 Hz and sampling rates are in the order of 2-10 kHz: The results shown later in this section are consistent with this assumption. With this in mind, only digital control within the context of the continuous design approach will be discussed here (for the alternative digital design approach see [53][54][4]).

### 6.4.3 The Discrete Transfer Functions

The discrete transfer function  $D(z)$  is defined as the ratio of the output sequence  $y(k)$  to the  $z$ -transform of the input sequence  $u(k)$ , i.e.

$$D(z) = \frac{Y(z)}{U(z)} \quad (6.3)$$

where  $k$  is the  $k^{\text{th}}$  data sample and the  $z$ -transform of a sequence  $x(k)$  is defined as [54]

$$X(z) = \sum_{k=-\infty}^{k=\infty} x_k z^{-k} \quad (6.4)$$

A continuous transfer function  $D(s)$  may be transformed to a digital transfer function  $D(z)$ , by a number of methods. The “exact” method is the substitution of

$$s = (1/T) \ln z \quad \text{i.e.} \quad z = e^{sT} \quad (6.5)$$

Where  $T$  is the sampling period and  $z^{-1}$  has the physical interpretation of a time delay of one period. However this substitution results in terms which are difficult to deal with and hence simpler approximations are used. These include numerical integration and matched pole zero methods. These will now be discussed.

#### 6.4.4 Creating a Digital Transfer Function From a Continuous Transfer Function Using Numerical Integration

The field of numerical integration of differential equations is quite complex, but the fundamental concept is to represent a given continuous transfer function  $D(s)$  as a differential equation and then to derive a difference equation whose solution is an approximation to the differential equation[54].

Analysis of a number of integration methods have resulted in substitution rules in which each occurrence of the complex frequency  $s$ , in  $D(s)$ , is replaced by an approximation containing the complex variable  $z$ , a number of these can be seen in Table 6.1

Method	Approximation
Forward Rectangular rule	$s \sim \frac{z-1}{T}$
Backward Rectangular rule	$s \sim \frac{z-1}{Tz}$
Trapezoid rule	$s \sim \frac{2}{T} \frac{z-1}{z+1}$

Table 6.1:  $s$  substitutions (in  $z$ ) corresponding to numerical integration.

The forward rectangular rule is simply Euler's integration method and the trapezoid rule is also known as the bilinear transform or Tustin's method [55]. The approximations of Table 6.1 map the  $s$ -plane to the  $z$ -plane. Now, in the  $s$ -plane  $s = jw$  is the stability boundary and the mapping of this boundary, by the three rules, onto the  $z$ -plane can be seen in Figure 6.7.

The stable region of the  $z$ -plane lies within the unit circle and it can therefore be seen that the forward integration method may result in an unstable digital realisation of a stable continuous transfer function. Furthermore, since Tustin's method maps the stable region of the  $s$ -plane exactly to the stable region of the  $z$ -plane it is the most commonly used of the approximations. An alternative approach to the creation of a digital transfer function from the continuous form is the matched pole zero method, this is now detailed.



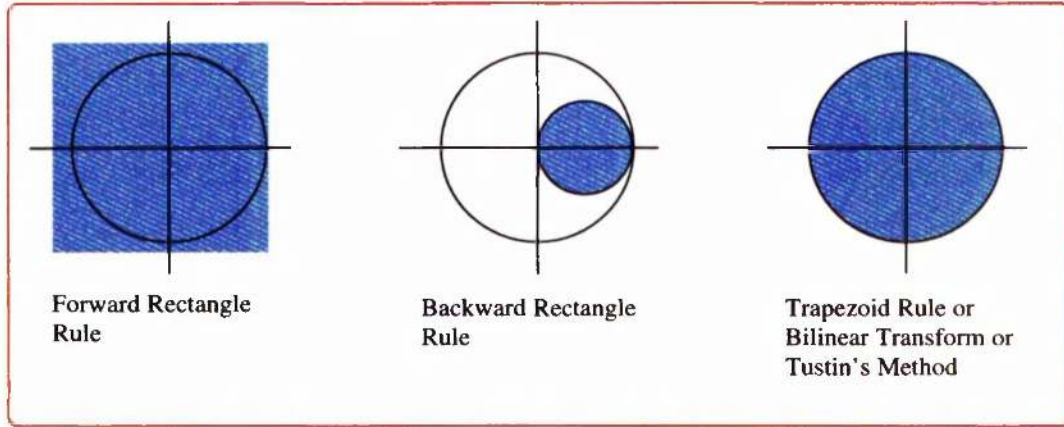


Figure 6.7: Mappings of the left half  $s$ -plane (stable region) to the  $z$ -plane via the rules of Table 6.1. Stable  $s$ -plane poles map into the shaded regions in the  $z$ -plane. The unit circle is shown for reference.

### Creating a Digital Transfer Function From a Continuous Transfer Function Using the Matched Pole Zero (MPZ) Method.

Since every point in the  $s$ -plane maps to a point in the  $z$ -plane through the mapping  $z = e^{sT}$  it should be possible to construct  $D(z)$  from  $D(s)$  by mapping the poles and zeros from one plane to the other, i.e. given a continuous zero-pole-gain representation

$$D(s) = K_s \frac{(s - a_1)(s - a_2) \cdots (s - a_n)}{(s - b_1)(s - b_2) \cdots (s - b_n)} \quad (6.6)$$

then

$$D(z) = K_z \frac{(z - c_1)(z - c_2) \cdots (z - c_n)}{(z - d_1)(z - d_2) \cdots (z - d_n)} \quad (6.7)$$

where  $c_i = e^{a_i T}$  and  $d_i = e^{b_i T}$ ,  $K_z$  is matched such that at DC, or at some low frequency, the gain of  $D(s)$  and  $D(z)$  are the same. Note, to find the DC gain of a digital transfer function simply substitute  $z = 1$  into  $D(z)$ . Further, it should be appreciated that as the sampling frequency increases (i.e the period decreases) the poles (and zeros) migrate towards the boundary of the unit circle.

## The “Delay Line Tap” Digital Implementation

Having designed a digital transfer function  $D(z)$  the next step is the actual implementation such that it can be realised by a computer algorithm.

Suppose  $D(z)$  is of the form

$$D(z) = \frac{Y(z)}{U(z)} \quad (6.8)$$

where the output  $Y(z)$  and input  $U(z)$  are both polynomials in  $z$ , then dividing the numerator and denominator of  $D(z)$  by the highest power in  $z$  yields

$$\frac{Y(z)}{U(z)} = \frac{a_0 + a_1 z^{-1} + \dots + a_{n-1} z^{-n+1} + a_n}{1 + b_1 z^{-1} + \dots + b_{n-1} z^{-n+1} + b_n z^{-n}} \quad (6.9)$$

and since  $z^{-1}$  is equivalent to a time delay of  $T$  seconds the expression of Equation 6.9 can be expressed in the time domain by writing  $z^{-n} y^*(t) = y^*(t - nT)$ <sup>1</sup> hence

$$\begin{aligned} y^*(t) = & a_0 u^*(t) + a_1 u^*(t - T) + \dots + a_n u^*(t - nT) \\ & - b_1 y^*(t - T) - b_2 y^*(t - 2T) \dots - b_n y^*(t - nT) \end{aligned}$$

This can be more succinctly written as

$$y_0 = a_0 u_0 + b_1 u_1 + \dots + a_n u_n - b_1 y_1 - b_2 y_2 \dots - b_n y_n \quad (6.10)$$

where the subscript 0 indicates the current value, the subscript 1 the first most recent past value, etc. In Table 18 a fragment of C program code is shown to demonstrate the implementation of a simple second order discrete transfer function. Obviously this is not complete and as such only details the relevant code.

It should be noted, that so far the order of  $D(z)$  has been proper (that is the order of the numerator and denominator are the same) and as such it is impossible, in practice, to implement exactly. This is because it requires the measurement of  $u_0$  and calculation of  $y_0$  to occur instantaneously. To overcome this, strictly proper (where the order of the denominator is at least one greater than the numerator) digital transfer functions should be designed.

<sup>1</sup>  $y^*(t) = \sum_{k=0}^{\infty} y(kT) \delta(t - kT)$ , that is  $y^*(t)$  is a sequence of unit impulses modulated by the function  $y(t)$  at the sampling instants  $t = T, 2T, 3T, \dots$

```

main()
{
    /* variable declarations etc here */

    while (time < end_of_time)
    {
        if new_sample()
        {
            u0 = input_adc();
            y0 = a0*u0 + a1*u1 + a2*u2 - b1*y1 - b2*y2;
            output_dac();

            /* now shift variables ready for the next sample */
            x2 = x1;
            x1 = x0;
            y2 = y1;
            y1 = y0;
            time = time+T
        }
    }
}

```

Table 6.2: A fragment of computer code for the implementation of a second order digital transfer function represented by the direct form [4]

The *form* or *structure* of Equation 6.10 is known as the direct form and can be represented graphically, as shown in Figure 6.8. Moreover, the direct form is not the only representation of a digital transfer function, others include the *canonic* and *delta* forms [56]. The significance of structure is that the order in which arithmetic operations are performed may result in different levels of quantization error [57]. Moreover, the coefficients  $a_n$  and  $b_n$  are also known as the *A* and *B taps* respectively. These digital implementations are commonly known as “delay line tap”.

Initially it was thought that a “delay line tap” controller implementation would provide a satisfactory basis for controller design. However, it soon became apparent that problems with this implementation fundamentally prevented its use. It was found that simple low order controllers could be satisfactorily implemented. However, as soon as the order of the controller increased to six or more, controllers suffered from errors. These errors arose due to the finite word length of the compensator coefficients and that as the number of poles and zeros increased so did the number of coefficients. Taking a continuous, zero pole gain, representation of a controller and converting this, via the bilinear transformation, to a set of



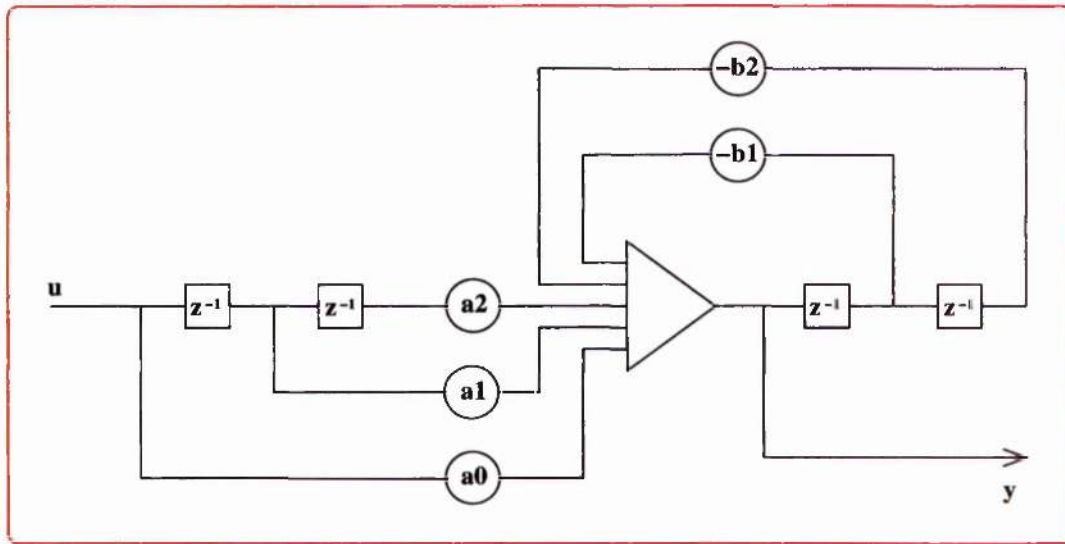


Figure 6.8: A graphical representation of the direct form of a second order digital transfer function

A and B taps resulted in poles very close to the unit circle<sup>2</sup>. Hence, due to the finite word length, and the number of A and B coefficients, the implemented controller poles migrated across the unit circle and thus produced an unstable controller. Hence the Euler integration method in conjunction with the state space representation were adopted as the basis for digital controller implementation, and is detailed in the next section.

### Digital controller implementation using the Euler integration method in conjunction with the state space representation

This method starts with the state space representation of the form:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{6.11}$$

---

<sup>2</sup>Transformations from continuous to digital representations were undertaken using Matlab or Octave built in functions

then taking the Laplace transform gives

$$\begin{aligned} sX &= AX + BU \\ Y &= CX + DU \end{aligned} \quad (6.12)$$

Where  $x \rightarrow X$ ,  $u \rightarrow U$  and  $y \rightarrow Y$  are the Laplace transforms of  $x$ ,  $u$  and  $y$  respectively and  $s$  is the complex frequency. Now, the complex  $s$  variable can be replaced by one of the integration approximations of Table 6.1 to produce a function in  $z$ ; i.e using the forward rectangle rule (Euler's method)

$$s \sim \frac{(z-1)}{T} \quad (6.13)$$

gives

$$\begin{aligned} \frac{z-1}{T}X &= AX + BU \\ Y &= CX + DU \end{aligned} \quad (6.14)$$

Now, remember that  $z$  corresponds to a forward shift of one time unit, that is,

$$zx(k) = x(k+1) \quad (6.15)$$

and hence in the time domain Equation 6.14 becomes the discrete equations:

$$\begin{aligned} x(k+1) - x(k) &= TAx(k) + TBu(k) \\ \text{i.e. } x(k+1) &= (I + TA)x(k) + TBu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \quad (6.16)$$

State space controllers, designed in this thesis, were implemented using dSPACE. dSPACE is a commercial set of hardware and software used to implement digital controllers. The software interfaces with Matlab, utilising the Real Time Workshop package. The hardware consists of 16 and 12 bit ADCs, and 12 bit DACs all with a voltage range of  $\pm 10$  volts. The digital signal processor (dsp) is a Texas instruments TMS 320C31. Further details and specifications can be found at the Internet site: [www.dspace.de](http://www.dspace.de)

To evaluate the suitability of digital controller implementation using dSPACE the analogue local controller (as detailed in Section 6.3) was used as a benchmark. The controller was

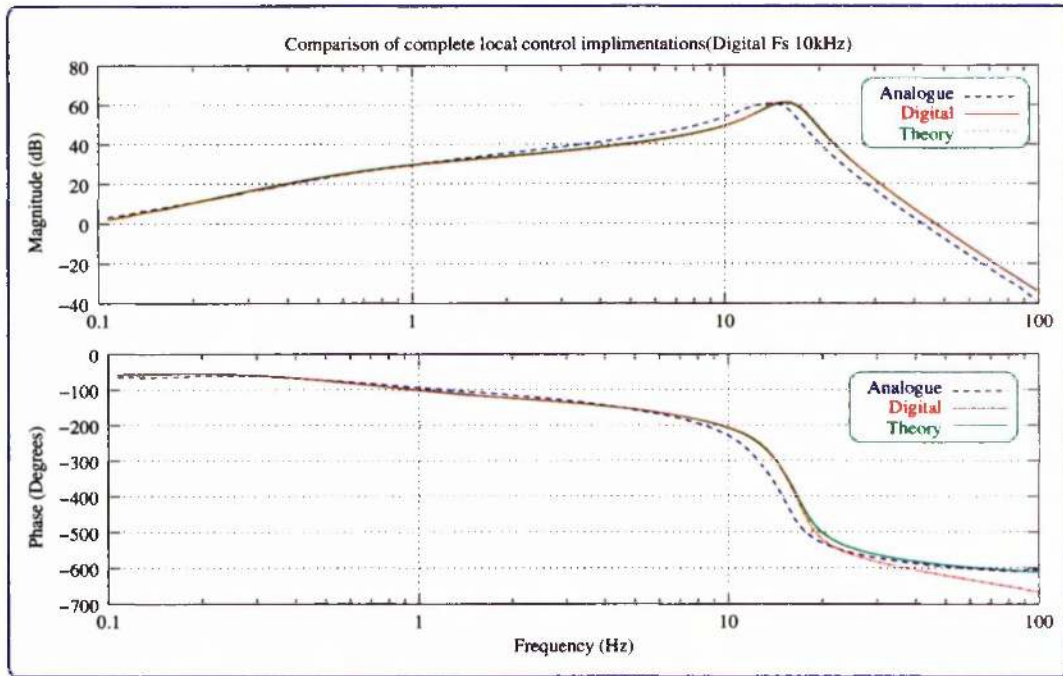


Figure 6.9: *Comparison of local control implementations*

represented in a state space format and implemented using the Euler integration method with a sampling frequency of 10 kHz. The digital and analogue responses of the system to a swept sine input were taken using a spectrum analyser. These were compared with theory and can be seen in the graph of Figure 6.9. As can be seen the digital response is consistent with theory. However, there is a slight difference in the analogue implementation. Here, the maximum gain, an artifact of the two resonant filters, is shifted. This arises due to known errors (tolerances) in the components used to construct the circuit. In particular the resonant frequencies of the two Scult ty filters are susceptible to variations in nominal capacitor and resistor values. Furthermore, it can be seen that the phase of the digital implementation is consistent with theory up to  $\sim 20$  Hz. There also exists a variable gain control for both the analogue and digital implementations.

Having shown that the digital implementation can satisfactorily reproduce the theoretical frequency response it was successfully used to damp motion of the double pendulum. An alternative approach to local controller design is the Model Based Observer [58] (MBO)

which can be designed in the bond graph domain and is presented in the next section.

## 6.5 Model Based Observer (MBO) Control

As the name suggests, a model based observer controller [58] incorporates a model of the physical system. Generally, before a controller is designed, knowledge of the physical system is required. This usually results in the generation of a model to facilitate system analysis. Having created this model, and assuming the model is an accurate representation of the physical system, it seems only natural to use the model as the basis of the controller. Moreover both the model and controller can be developed within the bond graph domain, thus providing a unified approach to model and controller design. In addition, these controllers are based in the physical domain and thus provide a more intuitive approach to controller design.

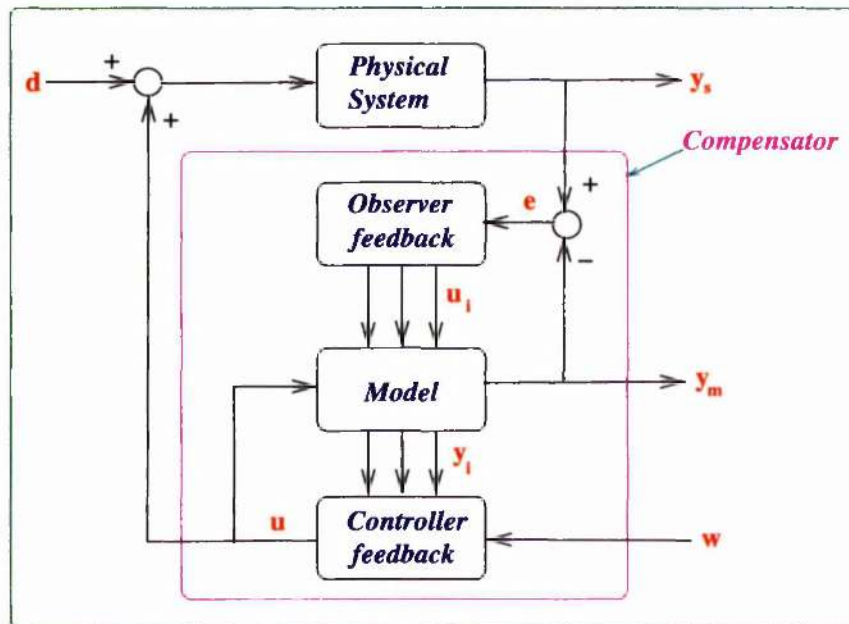


Figure 6.10: Schematic of a model based controller

A schematic of a model based controller can be seen in Figure 6.10. The following points should be noted.



- The block “system” represents the real physical system.
- The block “model” is a dynamic model of the physical system.
- The model has the same control inputs as the system.
- The output(s) of the system  $y_s$  and model  $y_m$  are compared to produce an error signal(s),  $e$ .
- The observer feedback is used to drive the states of the model to those of the system and hence drive the model error to zero.
- The block “Controller Feedback” contains a controller to drive the model, and hence the system, to a desired state.
- Both the model and compensator can be constructed in the bond graph domain.

A simple Mass-Spring system will now be used to demonstrate the principles of model based controller design in the bond graph domain. Having demonstrated the potential of this design methodology it will be extended to the more complicated double pendulum system. In each case the aim of the controller is to damp mass motion.

### 6.5.1 Mass-Spring Model Based Controller

A bond graph of a mass spring system is shown in Figure 6.11. In the following demonstration it will be used for both model and physical system purposes. Figure 6.12 is a hierarchical bond graph of the model based observer controller. Note:

- **Se:d** is a bond graph effort source which enables the injection of disturbances.
- **Sf:w** is a flow source used to provide the setpoint.
- **De:u** is a bond graph effort detector (sensor) which, in this instance, measures the control effort.
- **De:y** outputs the displacement of the mass.

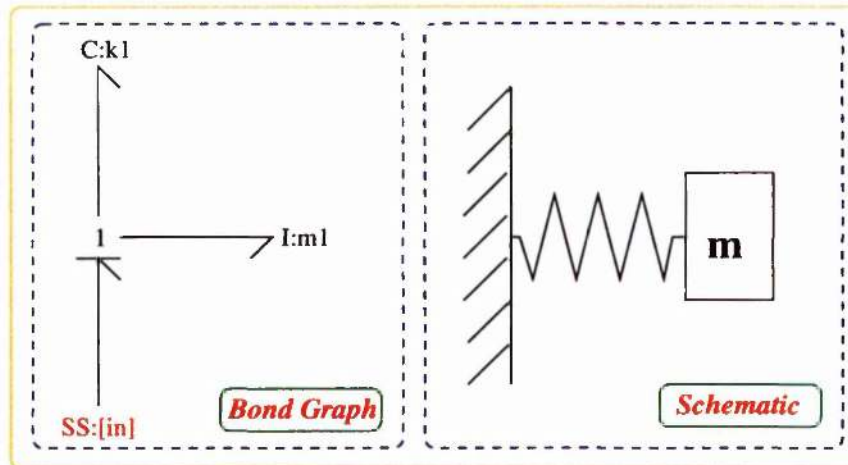


Figure 6.11: *Bond graph and schematic of a mass spring system*

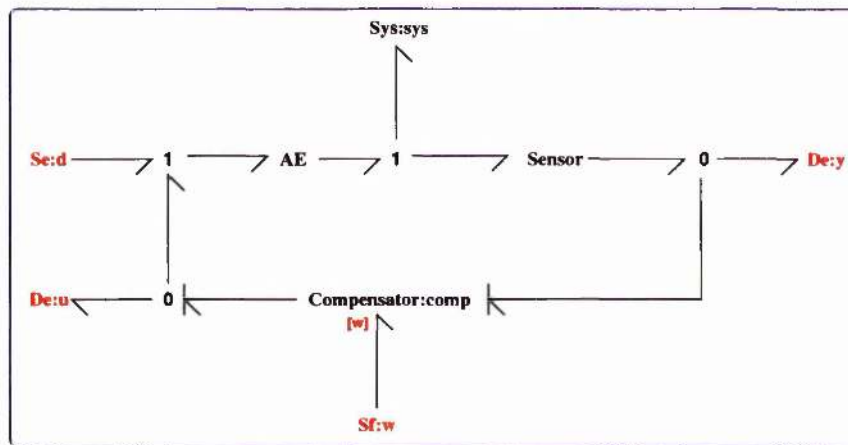


Figure 6.12: *Hierarchical bond graph of a model based controller*

- The bond graph subcomponent **Sensor** contains an **INTF** (integrate flow) component which is used to model a physical sensor that measures displacement.
- The subcomponent **Compensator** is a hierarchical bond graph which implements the compensator portion of Figure 6.10, i.e. incorporates the observer, the controller and the system model.





**SS:[w]** interfaces to the setpoint and as such provides an input (flow) and hence (via **R:r\_c**) imposes a force on both model and system.

Having constructed this set of bond graphs, MTT can be employed to create alternative model representations which in turn can be used to analyse the system. The structure file (mtt MBO struc view) can be seen in Table 6.3, giving the inputs, outputs, states and non-states (the non-state arises as a consequence of the derivative causality on the **C** component, within the **INTF** component, which is used to perform sensor inversion.

List of inputs for system MassSpring			
	Component	System	Repetition
1	u (disturbance)	MassSpring_d	1
2	u (set point)	MassSpring-w	1

List of nonstates for system MassSpring			
	Component	System	Repetition
1	mtt3	MassSpring_comp_mtt1	1

List of outputs for system MassSpring			
	Component	System	Repetition
1	model_disp	MassSpring_comp	1
2	y (control)	MassSpring-u	1
3	y (system_disp)	MassSpring-y	1

List of states for system MassSpring			
	Component	System	Repetition
1	k	MassSpring_comp_model	1
2	m	MassSpring_comp_model	1
3	mtt3	MassSpring_comp_mtt7	1
4	mtt3	MassSpring_sensor	1
5	k	MassSpring_sys	1
6	m	MassSpring_sys	1

Table 6.3: The structure files for the mass spring velocity damping model based observer

$$G = \begin{pmatrix} \frac{(r_o s)}{(k^2 + 2kms^2 + kr_c s + kr_o s + m^2 s^4 + mr_c s^3 + mr_o s^3 + r_c r_o s^2)} & \frac{r_c}{(k + ms^2 + r_c s)} \\ \frac{(-r_c r_o s^2)}{(k^2 + 2kms^2 + kr_c s + kr_o s + m^2 s^4 + mr_c s^3 + mr_o s^3 + r_c r_o s^2)} & \frac{(r_c (k + ms^2))}{(k + ms^2 + r_c s)} \\ \frac{(k + ms^2 + r_c s + r_o s)}{(k^2 + 2kms^2 + kr_c s + kr_o s + m^2 s^4 + mr_c s^3 + mr_o s^3 + r_c r_o s^2)} & \frac{r_c}{(k + ms^2 + r_c s)} \end{pmatrix} \quad (6.17)$$

The closed loop transfer functions  $G(s)$  of the MBO are detailed in the transfer function matrix of Equation 6.17, where  $r_o$  and  $r_c$  are the observer and controller damping coefficients respectively,  $m$  is the mass and  $k$  is the spring constant. Notice, that the transfer functions associated with the setpoint ( $G(i,2)$ ) do not contain terms involving the observer gains ( $r_o$ ), this is in accordance with the *separation principle* [59] which states that the observer and controller can be designed independently. Furthermore, the denominators of these transfer functions have the same form as the standard damped second order transfer function (Section 2.3.2) and therefore  $r_c$  can be chosen such that the closed looped system is critically damped, i.e.

$$r_c = 2\sqrt{(m \times k)} \quad (6.18)$$

Therefore, assuming the model can be driven to the same state as that of the physical system, (the task of the observer), it will also be critically damped. Impulsive inputs at **Se:d** enable an appropriate value for  $r_o$  to be chosen such that the model is driven to the same state as the system within an appropriate time scale. The value assigned to  $r_o$  is generally limited by sensor noise.

Notice that the overall bond graph structure is constructed such that both model and system are subcomponents, making this damping controller generic. Therefore should a system require a velocity damping controller, simply replace the system and model bond graphs, and reassign the values of the  $r_c$  and  $r_o$  coefficients. More complicated compensators can be designed in this manner. Thus creating a library of bond graph compensators.

(Simulated) step and impulse responses of this bond graph designed model based observer controller can be seen in Figures 6.14 and 6.15 respectively. Figure 6.14 shows the step response of both the “system” and model to a step input at the setpoint. The values of the coefficients,  $r_c$  and  $r_o$  were chosen such that,

$$r_c = 4\sqrt{(m \times k)} \quad (6.19)$$

and

$$r_o = 0$$



In Figure 6.15 both the “system” and model responses to an impulsive disturbance at the mass can be seen. Here the value of  $r_c$  is the same as earlier and  $r_o = 4r_c$ . These graphs suggest that the methodology of model based observer control has the potential to adequately control physical systems. In the next section the methodology will be used to design a local controller for a real physical multi-stage pendulum.

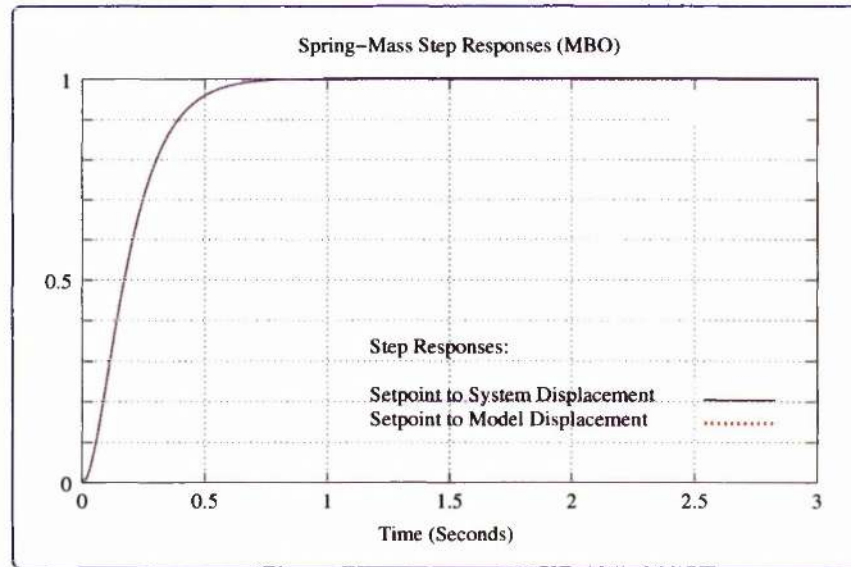


Figure 6.14: The response of the “system” and model to a step input at the setpoint

## 6.6 Local Control Design Using Bond Graphs

In this section a local controller, designed to damp the resonant modes of a double pendulum, is presented. The methodology is an extension of the previous section. The only difference is that the system to be controlled is a slightly more complicated multi-input multi-output (MIMO) system.

The bond graphs shown in Figures 6.16 through to Figure 6.18 are the set of hierarchical bond graphs used to produce a local controller for the double pendulum of Figure 6.1 (Page 109). Figure 6.16 is the top level bond graph. Notice it is of the same form as that used in the Mass-Spring MBO designed controller. However, here the bond graph compo-

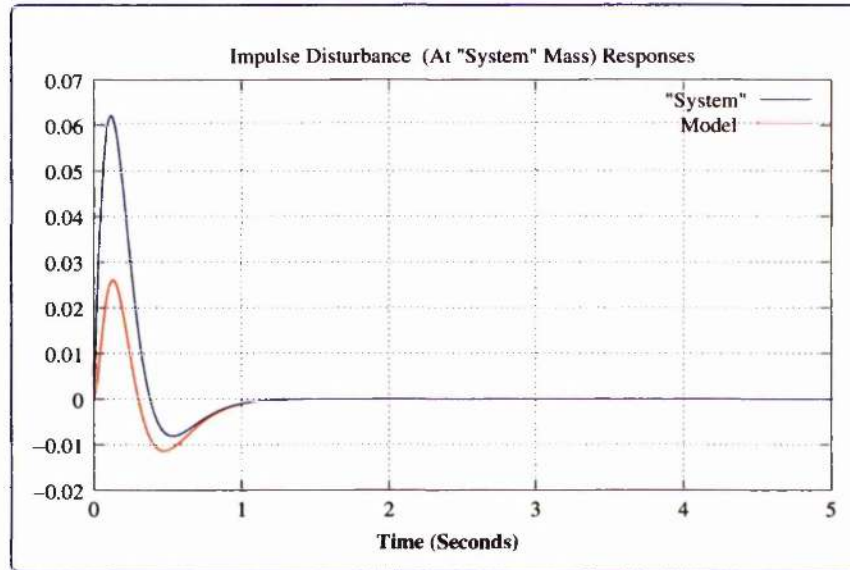


Figure 6.15: The response of the “system” and model to a impulsive disturbance of the mass

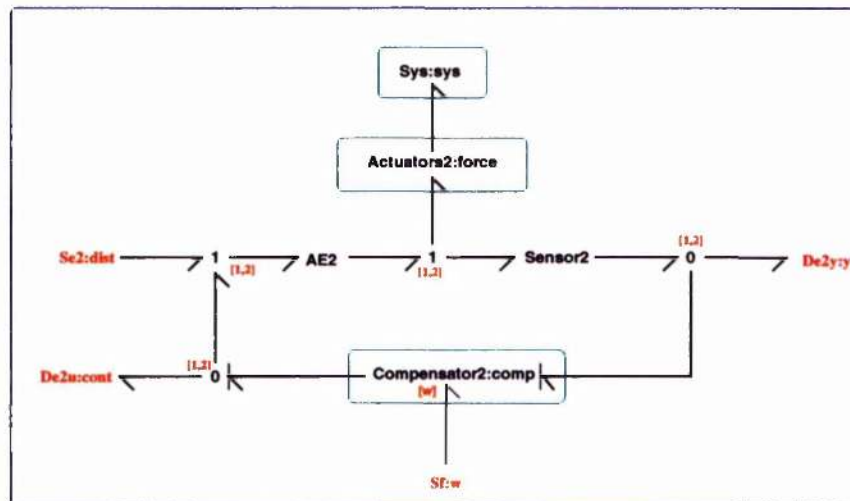


Figure 6.16: Top level MBO bond graph controller

ment **Actuators2** has been included to define the location of the two actuator/sensor pairs which provide longitudinal and tilt actuation forces. Moreover, bonds are now vector bonds which reflects the fact that these bonds form interconnections between components that contain two instances of the corresponding components used within the controller design of

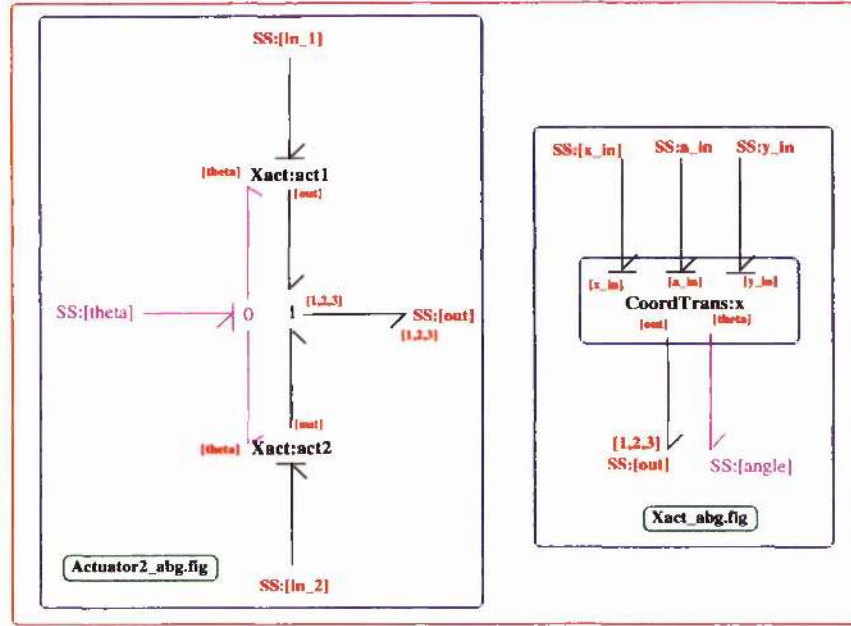


Figure 6.17: Bond graphs - *Actuators2* and associated subcomponent *Xact*.

Mass-Spring (Section 6.5.1): For example, the component **SE:dist**, as used in the Mass-Spring system, has changed to component **SE2:dist** which simply reflects the fact that this component contains two instances of the effort source **SE**.

The bond graph component **Actuators2** and its subcomponents can be seen in Figure 6.17. These bond graphs simply define the location of sensor/actuator pairs. Each subcomponent **Xact** provides the location of one collocated source/sensor. Here the single input to **Xact** carries the effort variable corresponding to the actuator force. Since an actuator only provides forces in the X direction, **Xact** has two **SS** components (**SS:a\_in** and **SS:y\_in**), with zero effort causality, these in turn connect to the Y and  $\Theta$  coordinates of the component **CoordTrans**. As detailed in Section 4.5.3 (Page 63), **CoordTrans** simply provides the correct geometric transformation between a point in an extended body and its centre of mass. Here actuation force applied in the X direction, at the location of an actuator, is mapped to force at the centre of mass and torque about the centre of mass.

The subcomponent **Compensator2** of Figure 6.18, contains two instances of the subcomponent **ContObs**. These subcomponents (see `ContObs_abg.fig` of Figure 6.18) are similar



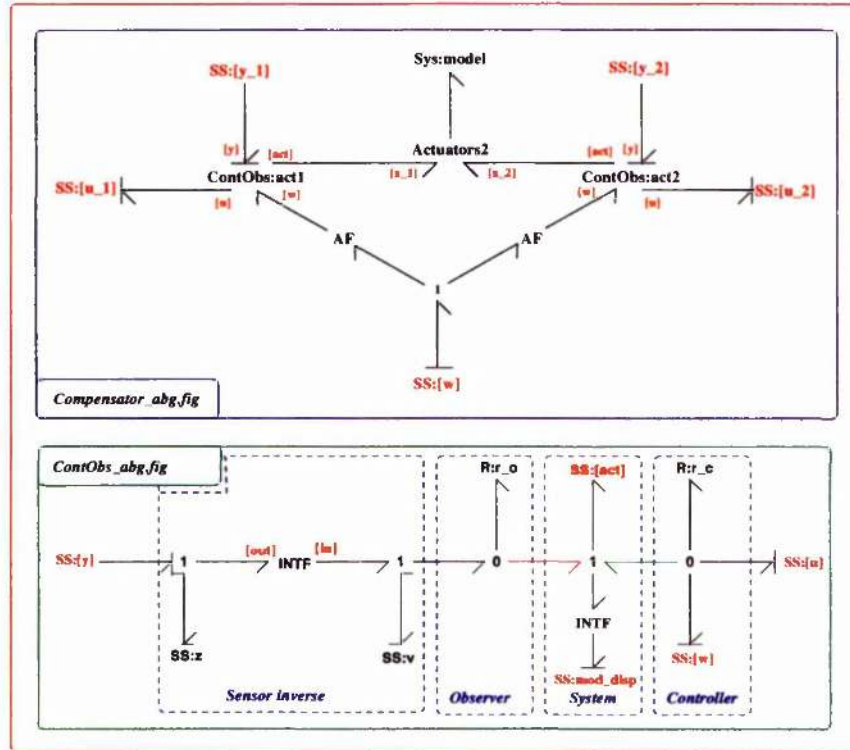


Figure 6.18: Bond graphs - *Compensator2* and subcomponent *ContObs*

to the compensator components as used in the Mass-Spring system. The only difference is that the component **Sys:mod** has been replaced by the port **SS:[act]** which allows the two **ContObs** components to interact with a single bond graph model of the physical system.

The bond graph model of the system, i.e. the double pendulum, is the same as that introduced earlier in this thesis (Chapter 4). Here, because the controller is designed to damp intermediate mass motion, well below the resonant frequencies of the suspension wires (violin modes), the simplest wire model is used. This models restoring forces due to linear extension and results in a model of order 14.

This completes the description of the bond graphs used to create a new double pendulum local controller. Notice, the basic structure of the controller is the same as that for the Mass-Spring system and that, in general, earlier bond graphs are simply re-used with little or no modification.

Again, using MTT, these bond graphs can be transformed into other system representations for system analysis purposes. Primarily, a state space representation can be generated and then by varying  $\mathbf{r}_c$  and  $\mathbf{r}_o$ , the system response, to step and impulsive inputs, can be analysed. Suitable values of  $\mathbf{r}_c$  and  $\mathbf{r}_o$  can be chosen such that the controlled system behaves in the desired manner. The compensator can then be extracted and implemented on the physical system. This process was undertaken, and a model based controller for the double pendulum produced. This was implemented using a state space representation, Euler integration and a sampling frequency of 2 kHz. The results of this compensator implementation are discussed in the next section.

## 6.7 Results

To compare controllers, an impulse was applied to the intermediate mass and the corresponding system ringdown measured using the sensors (detailed in Section 6.2.1). The sensor reading was amplified and the time sequence recorded using dSPACE. Typical impulse responses are shown in Figures 6.19 (Page 135) and 6.20 (Page 136), these contain time sequence plots, of both the original classically designed analogue controller and the new digitally implemented model based observer design, for channel 3 and channel 4 respectively. These two channels correspond to the two collocated sensor/actuators that measure longitudinal and tilt motion (see Figure 6.1 – Page 109). As can be seen, the normalised response of each controller implementation performs equally well with very similar ringdown periods. So the question arises as to why bother with yet another method that results in the same outcome?

Firstly the modelling of both the physical system and compensator can both be constructed in the bond graph domain: Thus providing a unified approach to model and controller design. Furthermore, the model based observer design provides the control engineer with a method of designing controllers in a more intuitive manor. For example: A physical system requires velocity damping. Hence, the system model within the compensator is connected to a dissipative component (i.e. a bond graph  $\mathbf{R}$  component) and “energy” is dissipated from the system model and hence the physical system is also damped. This does assume

the model states can be driven to the system states, the task of the observer, and thus requires a good model of the system. Another advantage is that the compensator is passive<sup>3</sup>, it is therefore assured to be stable. Furthermore, in this model based observer design, there are only two “tuning” parameters, ( $r_o$ ,  $r_c$ ), each of which is not simply some weighting factor, but instead have an intuitive meaning. For example, increases in the value of  $r_c$  increases the damping provided by the actuator, and increases in the value of  $r_o$  drives the modal states to the system states faster. Moreover, should the control engineer design a new compensator then a digital implementation is, in general, easier to update than the analogue equivalent.

## 6.8 Conclusions

In this chapter it was shown that a digital implementation of the original analogue controller was feasible. The Model Based Observer (MBO) controller design was then introduced, and its potential demonstrated through the design of MBO controller to damp the motion of a Mass-Spring system. The methodology was then extended to produce a MBO controller to damp the modes of a double pendulum. This new controller design was successfully implemented on a real physical system and its performance compared with the current analogue local controller. Both implementations were seen to have a similar performance. However, since the MBO design is based in the physical bond graph domain its design process is demonstrably more intuitive. Moreover, it was shown that the bond graph domain provides an ideal platform for the design of model based observer controllers and thus provides a unified approach to model and controller design.

---

<sup>3</sup>Note: It is also possible to produce active compensations by added active bonds

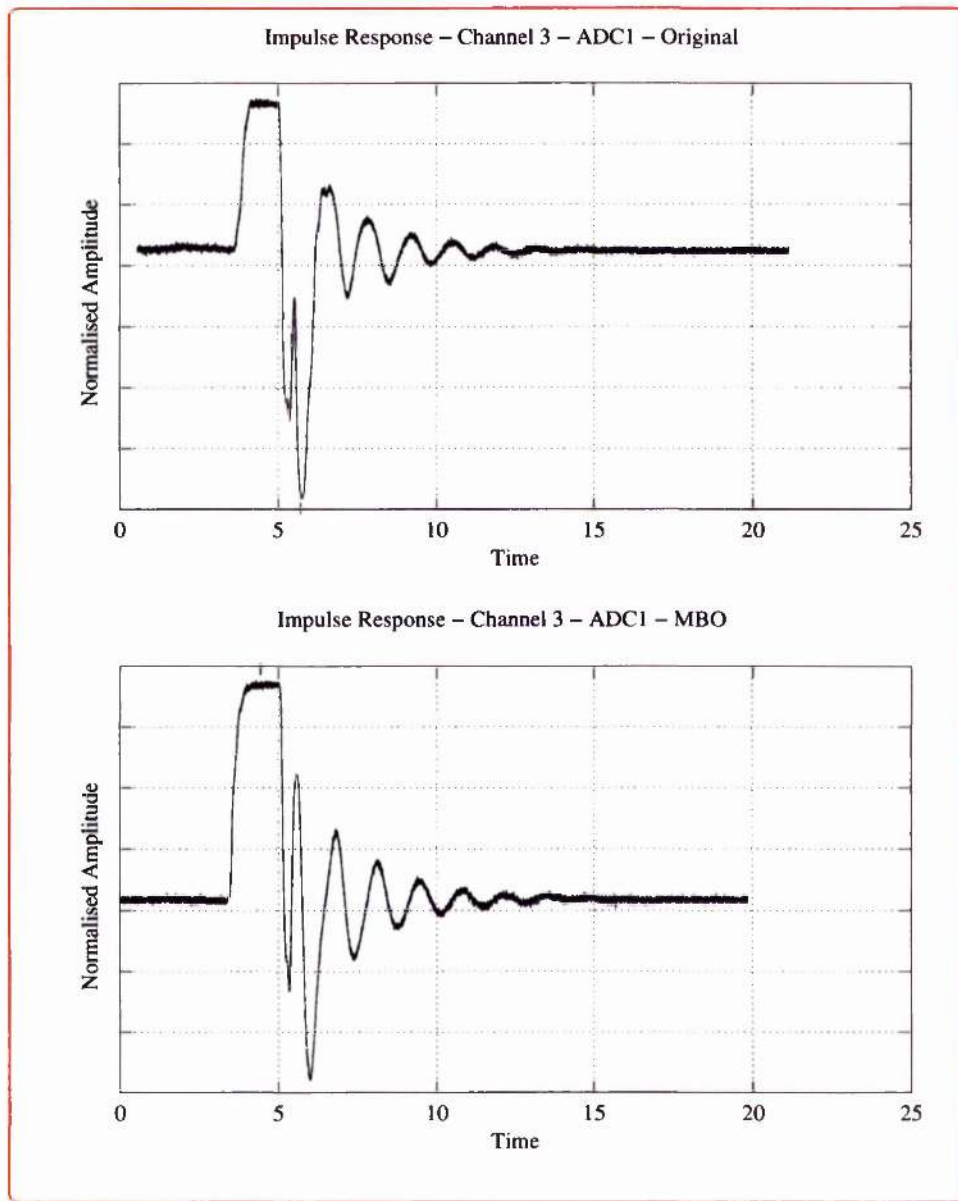


Figure 6.19: System response to impulse at intermediate mass - channel 3 (Top -> original, Bottom -> MBO)



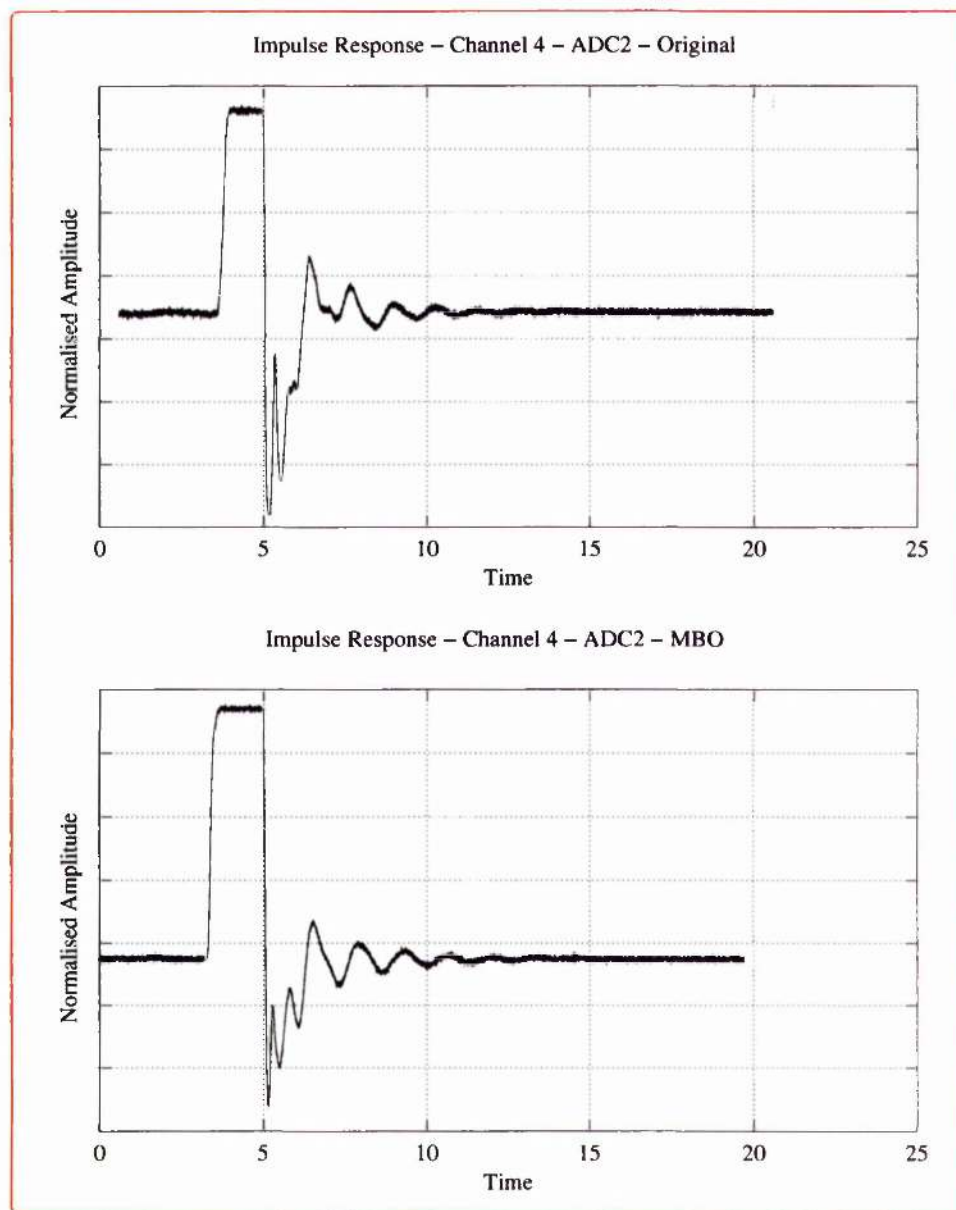


Figure 6.20: System response to impulse at intermediate mass - channel 4 (Top -> original, Bottom -> MBO)

## Chapter 7

# Global Control

“The problem with simulations is that they are doomed to succeed”

anon (but no doubt wise)

Whilst in operation the interferometric gravitational wave detector is maintained on a dark fringe. In this configuration noise, due to fluctuations in input light intensity, does not appear in the output signal. However, an offset from the dark fringe, by an amount  $\Delta x$ , results in laser beam intensity noise appearing in the output signal as

$$\delta x = \Delta x \frac{\tilde{I}}{I} \quad (7.1)$$

where  $\tilde{I}$  is the intensity noise of the light. It is anticipated that the laser light intensity, for GEO 600, will be stabilised to 1 part in  $\sim 10^7$  [60]. Thus to achieve a displacement sensitivity of  $10^{-20} \text{ m}/\sqrt{\text{Hz}}$  requires the root mean squared motion of the test mass to be less than  $10^{-13} \text{ m}$ . The displacement noise of the test mass, due to seismic noise, is  $\sim 10^{-6} \text{ m}$  at 1 Hz, and even with local control the damping of this motion is only reduced by a factor of 10 and so a servo gain of at least  $\sim 10^6$  is required.



## 7.1 Feedback to the Test Mass

To have a single loop with a gain of  $\sim 10^6$  would result in a minimum bandwidth of  $\sim 1$  kHz. Moreover, as a consequence of applying this level of control directly to the test mass there is no filtering of the control noise, and therefore actuators with exceptionally low noise characteristics are required. In addition, the use of coil magnet actuators is prohibited because attaching magnets to a test mass introduces significant mechanical loss and hence is a potential source of thermal noise.

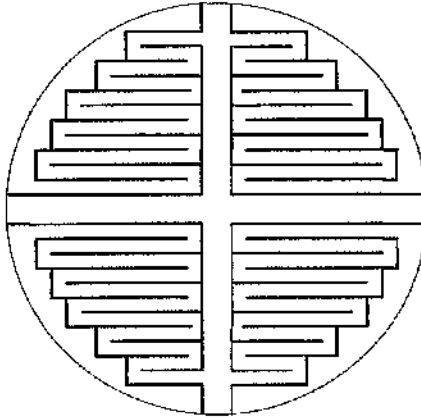


Figure 7.1: *One possible electrostatic drive configuration*

To overcome these difficulties it has been proposed that actuation at the test mass be provided by an electrostatic drive. This utilises the dielectric property of the test mass (a dielectric material is polarised by, and attracted towards, an electric field). A typical electrostatic drive can be seen in Figure 7.1, it consists of interlaced “combs” which can be deposited on a glass plate which in turn is positioned  $\sim 1$  mm behind the test mass. A bias voltage is applied across the combs, attracting the test mass towards the plate. The position of the test mass can then be modulated by modulating the bias voltage. Unfortunately the force generated by the electric fringe field of an electrostatic drive is limited by the breakdown voltage that may be applied across the combs and is generally much less than the force generated by coil magnet actuators. However, if the lower frequency/greater displacement actuation were to be applied at the intermediate mass, or at some higher stage in a chain of pendula, then the required actuation forces at the test mass can be reduced. Moreover,

since the low frequency actuation is to be applied at the intermediate (or higher) mass, coil magnet actuators can be used. The topology of this control configuration can be seen in Figure 7.2 and is discussed in the next section.

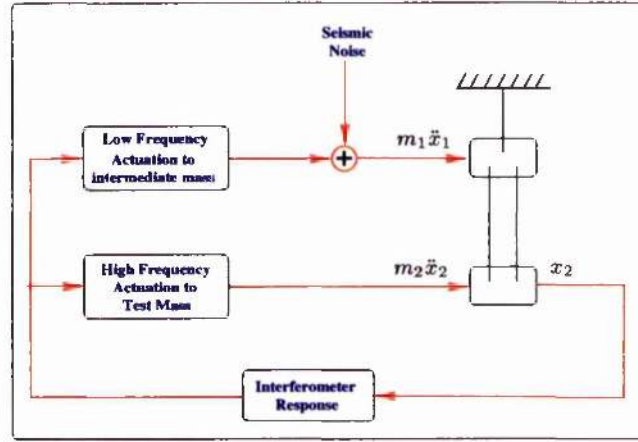


Figure 7.2: Schematic of split feedback topology

## 7.2 Split Feedback Control

To minimise the actuation range at the test mass, the crossover frequency, where feedback dominated by actuation at the intermediate mass changes to feedback dominated by actuation at the test mass, should be maximised. Since the feedback loop to the intermediate mass involves non-collocated sensing and actuation the elastic properties of the intervening material plays a more critical role in the response of the test mass. Furthermore since the bandwidth of the feedback loop to the intermediate mass is to be greater than that used in the local control design, the transverse modes of the suspension wires must be included (in pendulum models) to ensure that these modes are not excited, and thus create an unstable control law. So the goal of the global control design is to create a stable split feedback controller which maximises the control effort at the intermediate mass (thus reducing the actuation ranges required at the test mass) without exciting the transverse modes of the suspension wires, and *still* achieve a gain  $\sim 10^6$  at 1 Hz.

The rest of this chapter explores the possibility of using a physical model based observer to design the split feedback control for multi-stage pendula. The performance of this bond graph designed controller will then be assessed to see whether it meets the above design goals.

### 7.3 Split Feedback Controller Design Using A Bond Graph Model Based Observer (MBO)

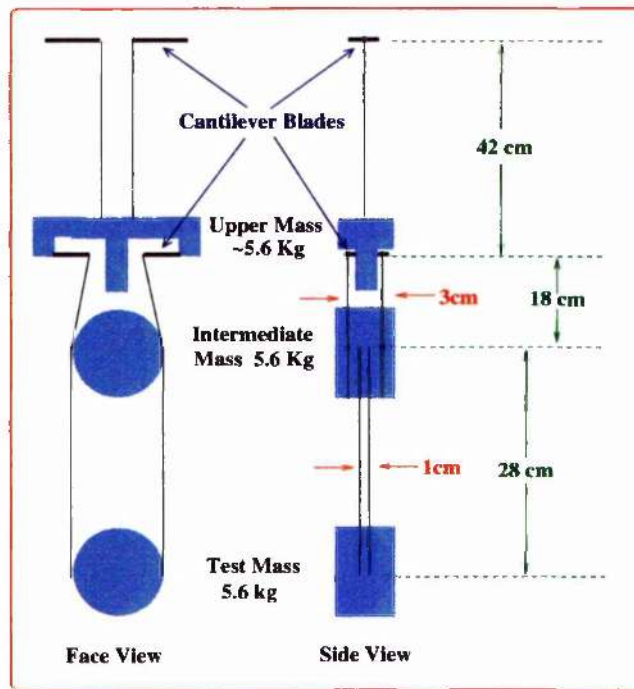


Figure 7.3: Schematic of a GEO 600 gravitational wave detector suspension. Notice that this is a three stage pendulum with two stages of cantilever blades.

The pendulum model, used in the design of the MBO split feedback controller, is a three stage pendulum rather than the two stage pendulum of the local control design<sup>1 2</sup> (Chap-

<sup>1</sup>The bond graph model of the double pendulum, as used in the local controller design, models the real physical pendula used to suspend mode cleaner optics: The mode cleaner is used to stabilise the beam geometry of the input laser [25][26]

<sup>2</sup>Initially a split feedback design was undertaken for a double pendulum. The move to a triple pendulum

ter 6). The bond graph triple pendulum is a model of a real physical triple pendulum used to suspend a main optic of GEO 600. A diagram of this triple pendulum can be seen in Figure 7.3 and the equivalent bond graph is shown in Figure 7.18 (Page 157). Note that, included in this system are two sets of cantilever springs. In accordance with the analysis of Husman [33] and Torrie [30] these cantilevers are modelled as simple springs, i.e bond graph C components, where the spring constant is given by the uncoupled mode frequency (umf). For a set of cantilevers, at a given pendulum stage, the uncoupled mode frequency is the resonant frequency of the cantilever blades supporting the mass of that stage only. For the upper set of blades (2 in number) the mode frequency is 2.57 Hz, the mass is 5.7600 Kg and hence  $k_{umf} = 739 \text{ N/m}$ , and for the lower blades (4 in number) the mode frequency is 3 Hz, the mass is 5.6034 Kg and hence  $k_{umf} = 498 \text{ N/m}$ .

## 7.4 Split Feedback MBO Bond Graphs

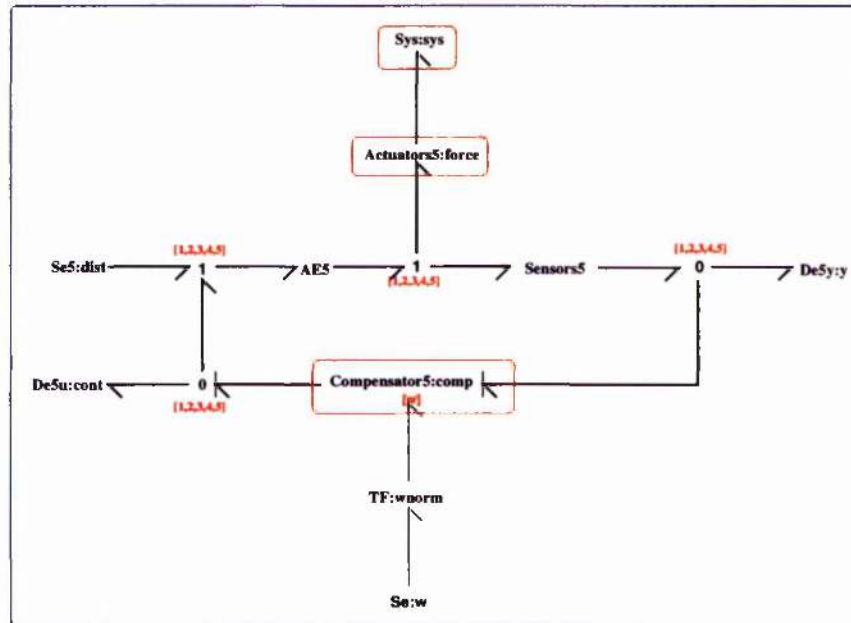


Figure 7.4: Top level hierarchical bond graph of a model based observer as used to design a split feedback global controller

was motivated by the extra degree of freedom (control) provided by the extra stage.



In this section the hierarchical MBO bond graphs, as used in the design of a split feedback controller, are presented.

The top level bond graph of a hierarchical model based observer (MBO) can be seen in Figure 7.4. The design of a global controller, like the local controller, is a bond graph designed MBO and hence many of the components used in the creation of the local controller can simply be re-used with little or no modification. For example, the top level bond graphs, of both the global controller (Figure 7.4) and local controller (Figure 6.16 Page 130), have the same structure. However, since there are now five actuator/sensor pairs (two actuators at the upper mass, two actuators at the intermediate mass and one at the test mass) the components **SE2**, **Sensor2**, **AE2** and **De2** (all described in Section 6.6) have been replaced with components where the suffix has changed from 2 to 5, which simply reflects the fact that these components now contain five instances of the basic components instead of two. Also the vector bonds connecting these components consists of five bonds rather than two. The main changes occur in the compound component **Compensator5** which will be discussed after the component **Actuator5** has had its properties detailed.

## 7.5 The Bond Graph Component Actuator5

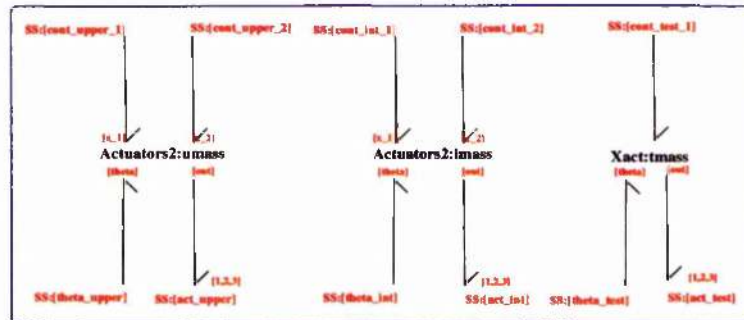


Figure 7.5: Bond Graph Component **Actuators5**: Defines the location of the actuators, two at the upper mass, two at the intermediate mass, and one at the test mass.

The bond graph component modelling the location of the actuators (hierarchical component **Actuators5**) can be seen in Figure 7.5 and simply consists of two instances of the com-

ponent **Actuators2**, which provide the location of actuators at the upper and intermediate masses, and a single instance of an **Xact** component, for modelling the actuator location at the test mass. These components were used in the local control design (see Section 6.6 (Page 129) for a full description of each component). Remember that the **Xact** component contains an instance of **CoordTrans** (Section 4.5.3) with zero forces applied to the input bonds associated with  $\Theta$  and  $Y$  coordinates, and hence provides actuation in the  $X$  direction only, i.e the axis of longitudinal pendulum motion. Furthermore, the two parameters,  $h$ , defining the horizontal offset from the centre of mass, and  $v$ , defining the vertical offset from the centre of mass, can be utilised to assess the effects of cross-coupling when the electrostatic drive is incorrectly aligned. That is, when perfectly aligned  $h = v = 0$  and  $v \neq 0$ ,  $h \neq 0$  otherwise. It should be noted that the **Actuator5** component simply defines the location of actuators; if actuator dynamics need to be modelled then extra components need to be created.

## 7.6 The Bond Graph Component Compensator<sup>5</sup> and Associated Subcomponents

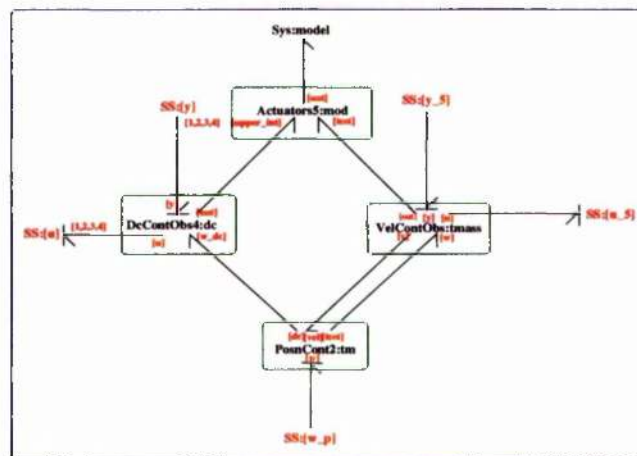


Figure 7.6: Bond graph component **Compensator5**: Used to produce a split feedback compensator.



The bond graph **Compensator5** can be seen in Figure 7.6. It consists of a number of subcomponents, each of which will now be detailed.

### 7.6.1 The Bond Graph Component Sys

This is a compound bond graph model of the physical system for which a compensator is to be designed. For the purpose of global control design this model is of a triple staged pendulum, where the suspension wires model both linear extension and transverse modes. These bond graph components are all detailed in Chapters 4 and 5.

### 7.6.2 The Bond Graph Component Actuators5

This component is described above (see Section 7.5)

### 7.6.3 The Bond Graph Component VelContObs

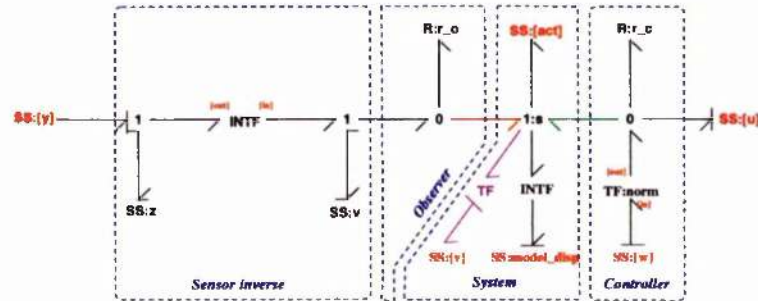


Figure 7.7: The bond graph of the component **VelContObs**. It extends the **ContObs** component by providing an extra port to convey the flow variable at the 1 junction 1:s to external components

The bond graph component **VelContObs** can be seen in Figure 7.7 and is simply a duplicate of the **ContObs** component (see Section 6.6) with the addition of the **SS:[v]** port, which is included to convey the flow variable from the 1 junction 1:s, to other bond graph components. When used as a subcomponent within **Compensator5** the port **SS:act** will impose,

upon junction **1:s**, the model's test mass velocity. The port **SS:[v]** can then convey this variable to the **PosnCont** component, where this velocity is integrated and subsequently used to provide actuation forces relating to the test mass position.

#### 7.6.4 The Bond Graph Component PosnCont2

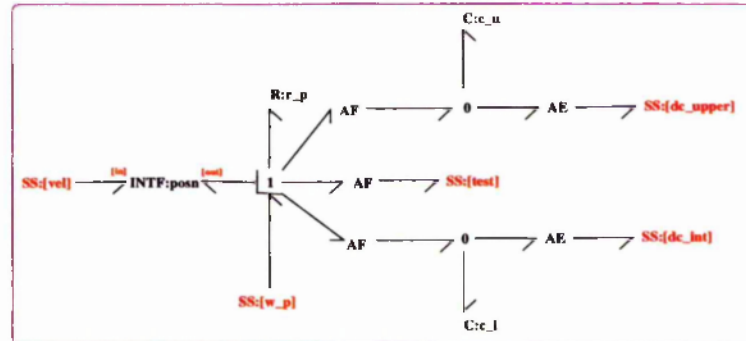


Figure 7.8: The *PosnCont2* component

The component **PosnCont2** (Figure 7.8) is used to provide restoring forces relating to the position of the test mass. Inputs to this component are test mass velocity (**SS:[vel]**) and set point (**SS:[w\_p]**). There are three outputs, one providing proportional action (**SS:[test]**) and two providing proportional-integral action (**SS:[dc\_upper]** and **SS:[dc\_int]**). For the purposes of global control the proportional action should be applied at the test mass and proportional-integral should be applied at the upper and intermediate masses. For simplicity the four port component **PosnCont** (Figure 7.9) will be used to illustrate the ideas of a position control bond graph component. The component **PosnCont2** (Figure 7.8) simply provides two ports with proportional-integral action instead of one.

#### A Description Of The Component PosnCont

In Figure 7.9 the annotated bond graph component **PosnCont** can be seen. This component takes a velocity ( $f_v$ ) (input at the port **SS:[vel]**) and then converts this to displacement via the component **INTF** (Section 3.7, Page 41). This displacement variable ( $e_d$ ) is carried by

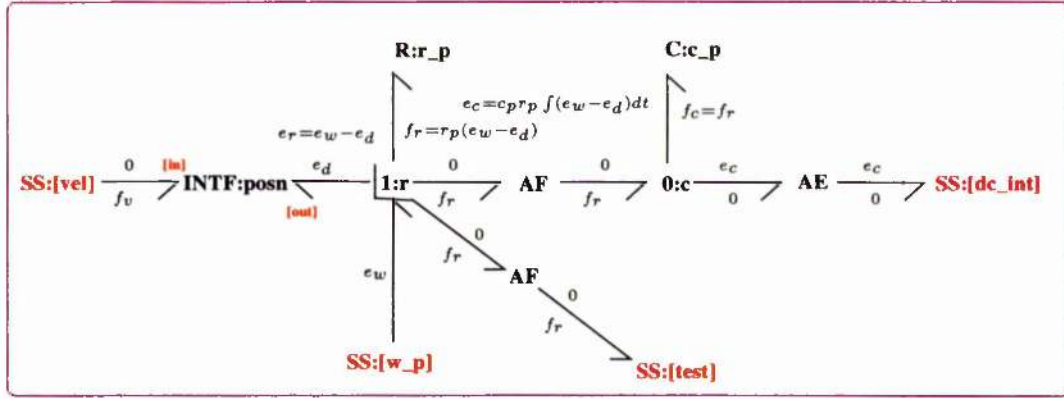


Figure 7.9: Bond graph component **PosnCont**, provides proportional and integral action for position control

the effort variable of the **INTF** output and hence, when the output bond of this component is attached to the **1** junction **1:r**, the input ( $e_R$ ) to the resistive component **R:r\_p** is given by

$$e_R = e_w - e_d \quad (7.2)$$

where  $e_w$  is the effort variable associated with the **SS** component **SS:[w\_p]** and is the set-point for test mass position control. The output flow ( $f_R$ ), of the **R** component, is given by

$$f_R = r_p(e_w - e_d) \quad (7.3)$$

and therefore this output can be used to provide a restoring force proportional to the error in the test mass' displacement and the setpoint. However, using only control forces proportional to the error results in a controller with a steady state error. This can be overcome through integration of the error signal ( $e_w - e_d$ ). The **C:c\_p** component attached to the **0** junction **0:c** can be used for this purpose. The input to this **C** component is  $r_p(e_w - e_d)$  and hence, because the **C** component has integral causality, the output is

$$e_c = c_p r_p \int (e_w - e_d) dt \quad (7.4)$$

and ensures zero steady state error. The inclusion of the **AE** and **AF** (Section 3.7) components simply ensure that only the variables of interest are propagated. The port **SS:[test]**

permits proportional action to be applied at the test mass whereas the port **SS:[dc\_int]** permits proportional–integral action to be applied at the intermediate mass. As stated above, the **PosnCont2** (Figure 7.8) simply has an additional proportional–integral action port, which is used to apply proportional integral action at the upper mass (port **SS:[dc\_upper]**).

### 7.6.5 The Bond Graph Components **DcContObs4**, **DcContObs2** and **DcContObs**

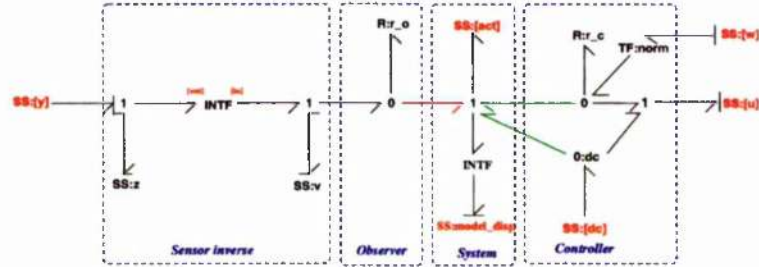


Figure 7.10: The bond graph component **DcContObs** provides an extra port c.f. **ContObs** for the inclusion of integral action forces associated with position control.

The bond graph component **DcContObs** can be seen in Figure 7.10. The only difference between this and the component **ContObs** (Section 6.6), is that an additional **SS** component (**SS:[dc]**) has been added which provides a port for the injection of proportional–integral action associated with the position control. This port has a bond connecting it to the **0** junction **O:dc** which is used to apply proportional–integral actuation forces to both the model and physical system. Since this is the only change to the original **ContObs** component it still retains all the features detailed earlier (Section 6.6), i.e it still provides velocity damping and observer gains.

The bond graph component **DcContObs2** (Figure 7.11) contains two instances of the component **DcContObs** (Figure 7.10). With reference to Figure 7.11 the various **SS** components should be connected to the following external bonds:

- **SS:[y\_1]** and **SS:[y\_2]** are associated with sensor outputs of the upper (intermediate) mass (of the “physical system”).



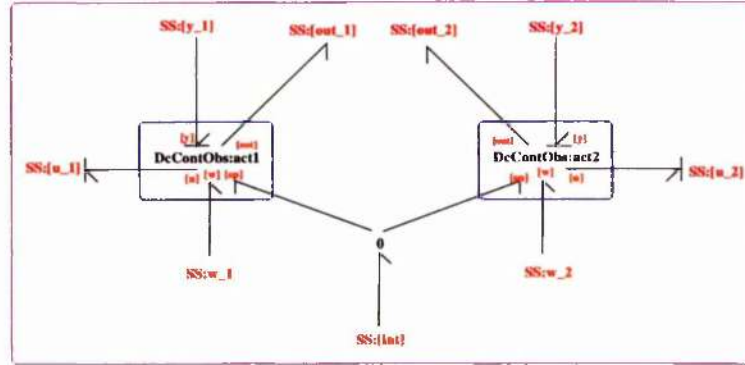


Figure 7.11: The bond graph component **DcContObs2** containing two instances of the component **DcContObs**

- **SS:[u\_1]** and **SS:[u\_2]** are associated with the generated controller forces for each of the upper (intermediate) mass actuators (of the “physical system”).
- **SS:[out\_1]** and **SS:[out\_2]** connects to the ports associated with the upper (intermediate) mass of subcomponent **Actuator5** of component **Compensator5** (i.e. to the system model not the “physical system”).
- **SS:[int]** connects to the port of **PosnCont** associated with the proportional–integral action and thus carries the position control forces.
- **SS:w\_1** and **SS:w\_2** provide individual set points for each of the upper (intermediate) mass actuators. Thus providing dc offsets.

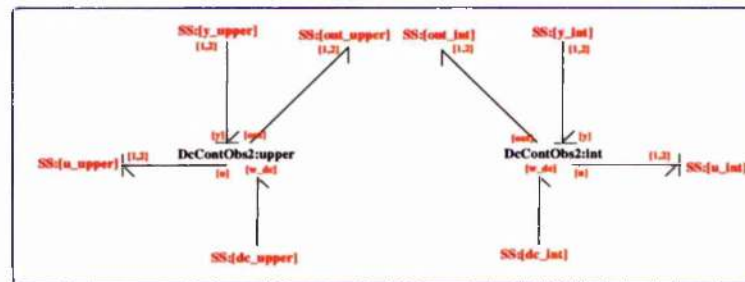


Figure 7.12: The bond graph component **DcContObs4** containing two instances of the component **DcContObs2**



The bond graph component **DcContObs4** (Figure 7.12) simply contains two instances of the **DcContObs2** component: **DcContObs2:upper** for the two actuators at the upper mass and **DcContObs2:int** for the two actuators at the intermediate mass.

### 7.6.6 Augmentation of the Observer

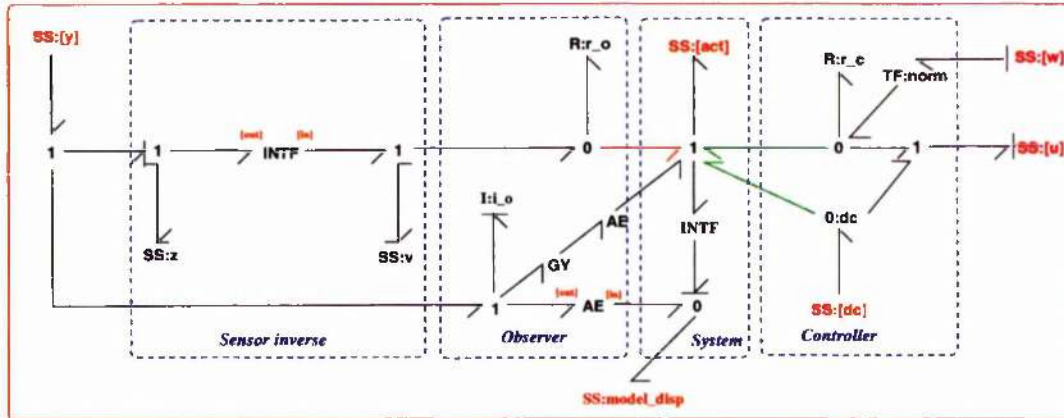


Figure 7.13: A modified **DcContObs** component incorporating integral observer action, which is used to drive the displacement error between system and model to zero

In the local control MBO design, the observer (used to drive the states of the model to those of the real physical system) simply consisted of an **R** component, which provided forces proportional to the error between system and model velocities. This was adequate for damping control, however this is not true for the global control and hence additional elements need to be provided. This can be achieved through the inclusion of an integrator (in this case an **I:i\_o** component) as shown in Figure 7.13. Notice that the input to this component is simply the error between the system and corresponding model displacements and thus, whilst the error is non zero, observer forces (integral action) will be provided to the model. The **GY** is simply used to convert the integrated error, associated with flow output of **I:i\_o**, from a flow to an effort variable and the **AE** components are used to ensure that only the desired variables are propagated. Naturally both the **DcContObs** and **VelContObs** components require this modification.

This completes the detailing of the subcomponents found within the hierarchical component

**Compensator5.** In the next section the performance of this bond graph designed MBO will be assessed. It should be noted that the state space representation of this component is precisely that which would be implemented in the digital controller software/hardware of dSpace (or equivalent).

## 7.7 Analysis of the Model Based Observer Split Feedback Controller

Having created the bond graph components of this MBO, the software program MTT was utilised to transform this core representation into the state space representation (this process is detailed in Section 4.7), Octave (similar to Matlab) was then used to carry out system analysis.

In the first instance, constraints on actuator forces are ignored, that is, it is assumed that action forces are not limited.

It should be appreciated that this is a very high order system: The triple pendulum model has thirty-six states, the complete MBO bond graph design has eighty-five states, and the compensator alone has forty-four states. With this in mind one might imagine that designing a split feedback controller would be a very difficult endeavour. However, as will be shown, using the MBO significantly reduces this difficulty. This is because there are only a small number of “tunable” parameters. Moreover, these parameters are not just some matrix weights or some such hidden weighting factors where physical insight is lost, but instead have a transparent physical interpretation. These parameters can be seen in Table 7.1, the first five parameters ( $r_{cu\_1}$ ,  $r_{cu\_2}$ ,  $r_{ci\_1}$ ,  $r_{ci\_2}$  and  $r_{ct\_1}$ ) are simply the damping coefficients associated with the local control. That is, this global control design also incorporates a local controller providing velocity damping. The remaining three parameters,  $r_p$ ,  $c_u$  and  $c_i$  are associated with the position control: The parameter  $r_p$  is the coefficient for the proportional control (i.e. provides the actuation force at the test mass) and the parameters  $c_u$  and  $c_i$  are the coefficients of the integral action (for upper mass intermediate masses respectively). Therefore, the effective split feedback design, for this thirty-six state multi-stage pendulum,

reduces to choosing appropriate values for the three parameters  $r_p$ ,  $c_u$  and  $c_i$ , but of course this assumes that the local control damping parameters have already been chosen.

To determine appropriate values for each of the damping coefficients all parameters were initially set to zero, then each set of damping coefficients, ( $r_{cu\_1}$  and  $r_{cu\_2}$  for the upper mass,  $r_{ci\_1}$  and  $r_{ci\_2}$  for the intermediate mass, and  $r_{ct\_1}$  for the test mass) were taken in turn, and appropriate numerical values assigned such that when a (simulated) impulse was applied, at respective masses, the ringdown was reduced to a minimum. Whilst determining these numerical values all observer gains were set to  $1 \times 10^5$ . Although these are not the values one would use in a real implementation it is valid here because they are simply being used to assist the design of the controller. Remember, in accordance with the separation principle, the controller and observer can be designed independently.

$r_{cu\_1}$	=	60.0;	%	upper mass damping coefficient (act 1)
$r_{cu\_2}$	=	60.0;	%	upper mass damping coefficient (act 2)
$r_{ci\_1}$	=	35.0;	%	int mass damping coefficient (act 3)
$r_{ci\_2}$	=	35.0;	%	int mass damping coefficient (act 4)
$r_{ct\_1}$	=	55.0;	%	test mass damping coefficient (act 5)
$r_p$	=	40;	%	proportional action
$c_i$	=	100;	%	integral action (intermediate mass)
$c_u$	=	100;	%	integral action (upper mass)

Table 7.1: A fragment of the *numpar* file (numerical parameter file) for the MBO split feedback controller

Having chosen the damping parameters, the remaining three split feedback parameters can be chosen. The value of  $r_p$  determines the actuation force at the test mass, reducing its value reduces the actuation force at the test mass but, for a given value of  $c_i$  and  $c_u$ , increases the time taken for the system to reach the steady state. The relative values of  $c_i$  and  $c_u$  determine the relative steady state actuation forces at the upper and intermediate masses. Naturally if one of these parameters is set to zero, steady state actuation forces will only be applied at the mass associated with the non zero  $c$  parameter. The complete table of parameters (i.e the *numpar* file for this system) can be found in Appendix C (Page 170).

In Figure 7.14 actuator forces, as a result of a simulated step input at the position setpoint ( $SS:[w\_p]$ ), can be seen, and the resulting displacements are shown in Figure 7.15. It should be noted that this MBO observer is a linearized representation of the non-linear MBO and



hence forces and displacement scale linearly. To investigate the damping qualities of the controller a simulated impulse was applied at the upper mass, the resulting motion of the upper, intermediate and test can be seen in the graphs of Figure 7.15. The bode plot of the compensator (**Compensator5**, (i.e that part of the design which would be implemented as a digital controller) can be seen in Figure 7.17

## 7.8 Conclusions And Comments

The graphs of Figures 7.14 and 7.15 shows that for both (simulated) step and impulse responses the system behaves exceptionally well. However, as mentioned earlier, this analysis has not considered constraints on actuation forces. Moreover, as can be seen in the bode plot of Figure 7.17, the split feedback dc gain at the upper and intermediate masses is very much less than the design goal of  $10^6$  (i.e 120 dB). Furthermore, this MBO design does not contain the necessary filtering to prevent sensor noise coupling into pendulum motion (work in progress). Yet, this very simple design process, based in the physical domain of bond graphs, and model based observers, has yielded a controller which, under the less severe design constraints, controls the (simulated) triple pendulum extremely well. Also, it should be remembered that only three parameters, which have a physically intuitive meaning, were used in the creation of the split feedback synthesis, and as such makes the design process that much simpler.

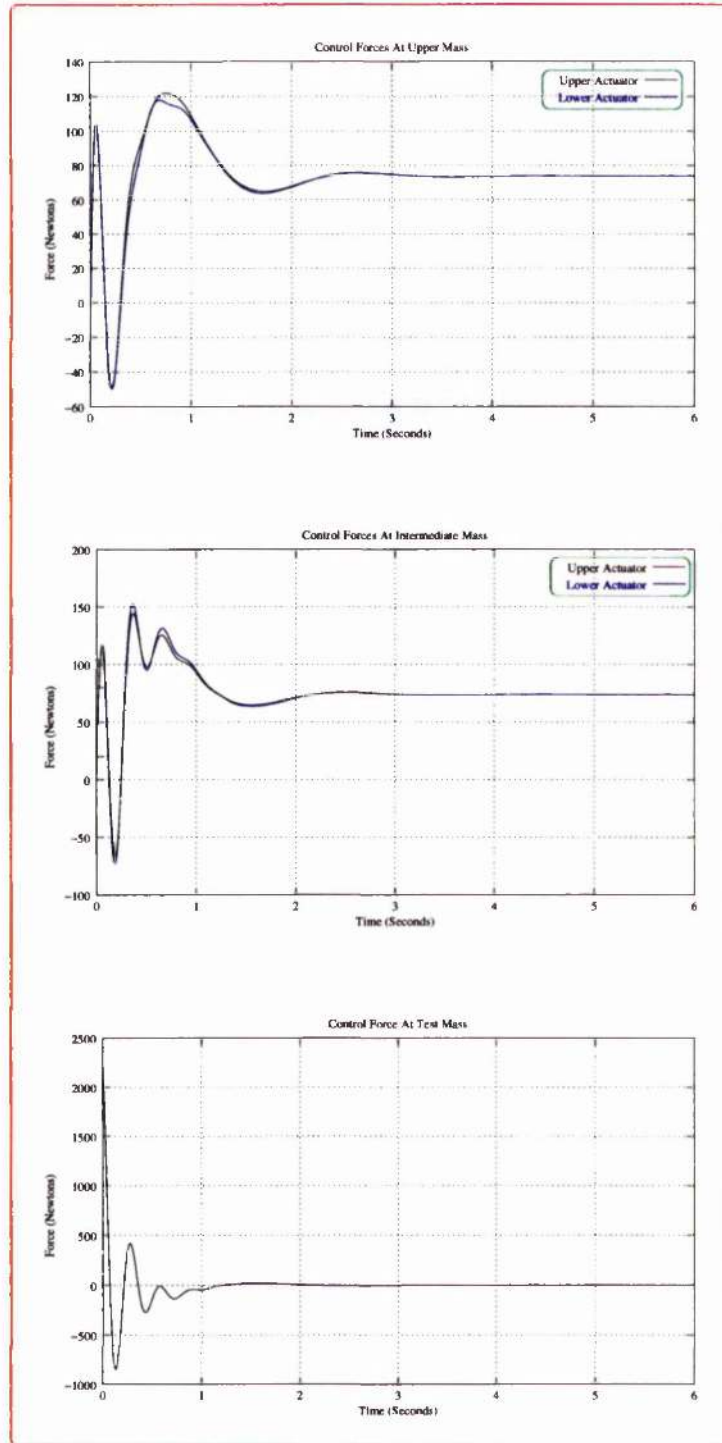


Figure 7.14: The actuation forces at the upper, intermediate and test masses in response to a step input at the test mass setpoint



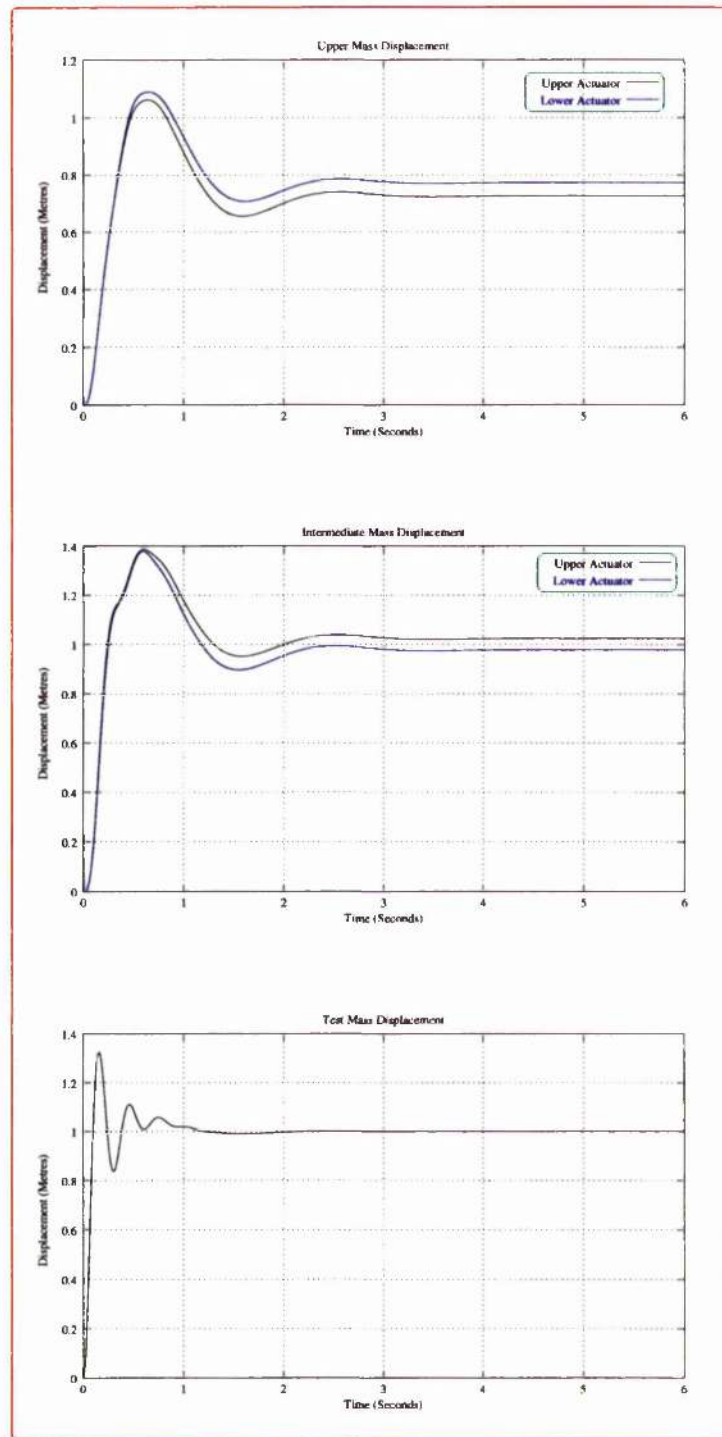


Figure 7.15: The resultant displacement of the upper, intermediate and test masses to a step input at the test mass

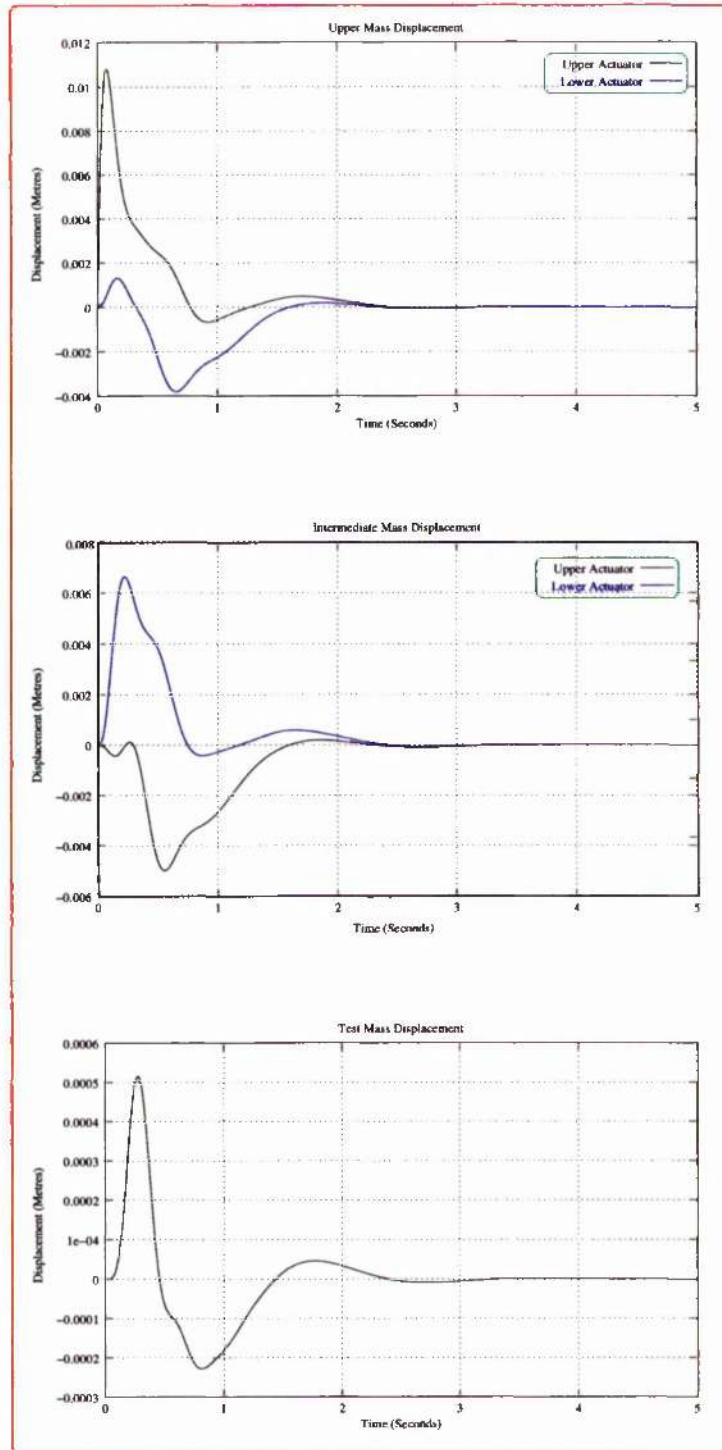


Figure 7.16: *The impulse response of the upper, intermediate and test masses - impulse applied at upper mass*

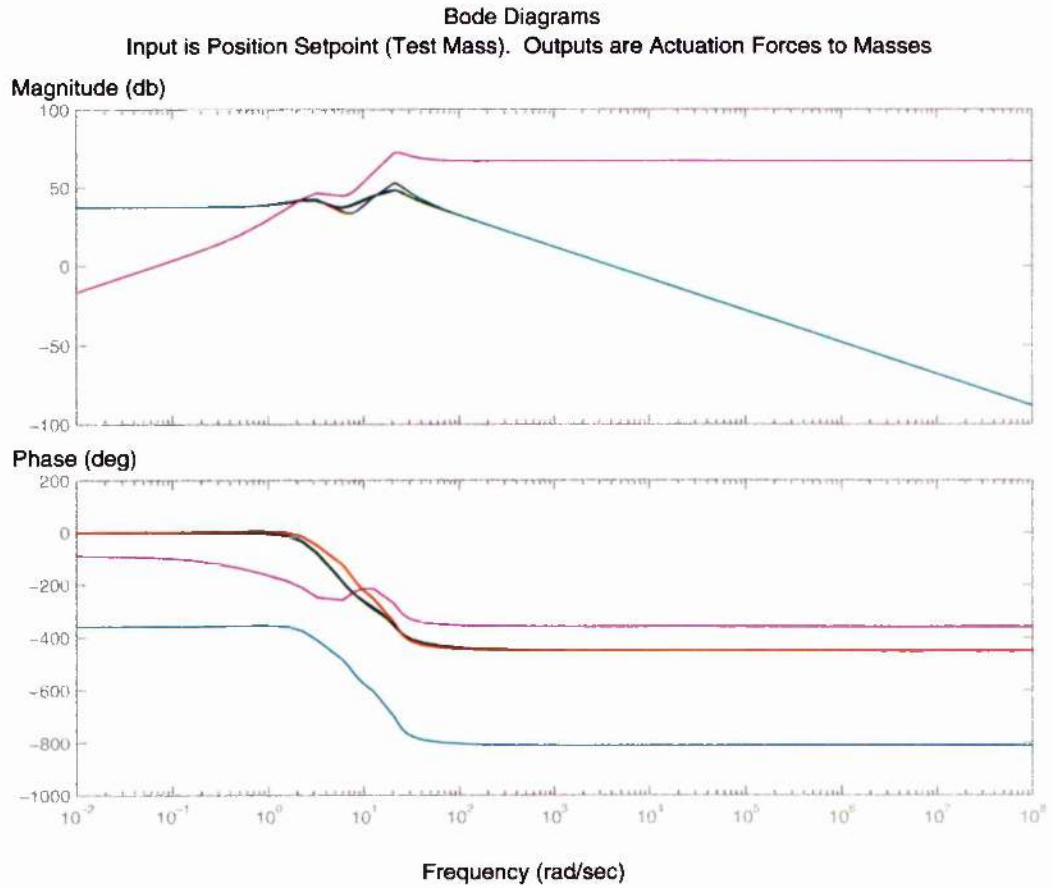


Figure 7.17: The bode plots: Input is position setpoint (test-mass) and outputs are actuator forces to masses: Magenta:- test mass, Other colours are those to the upper and intermediate masses. These all have approximately the same magnitude because the coefficients  $c_u$  and  $c_i$  have been assigned same values.

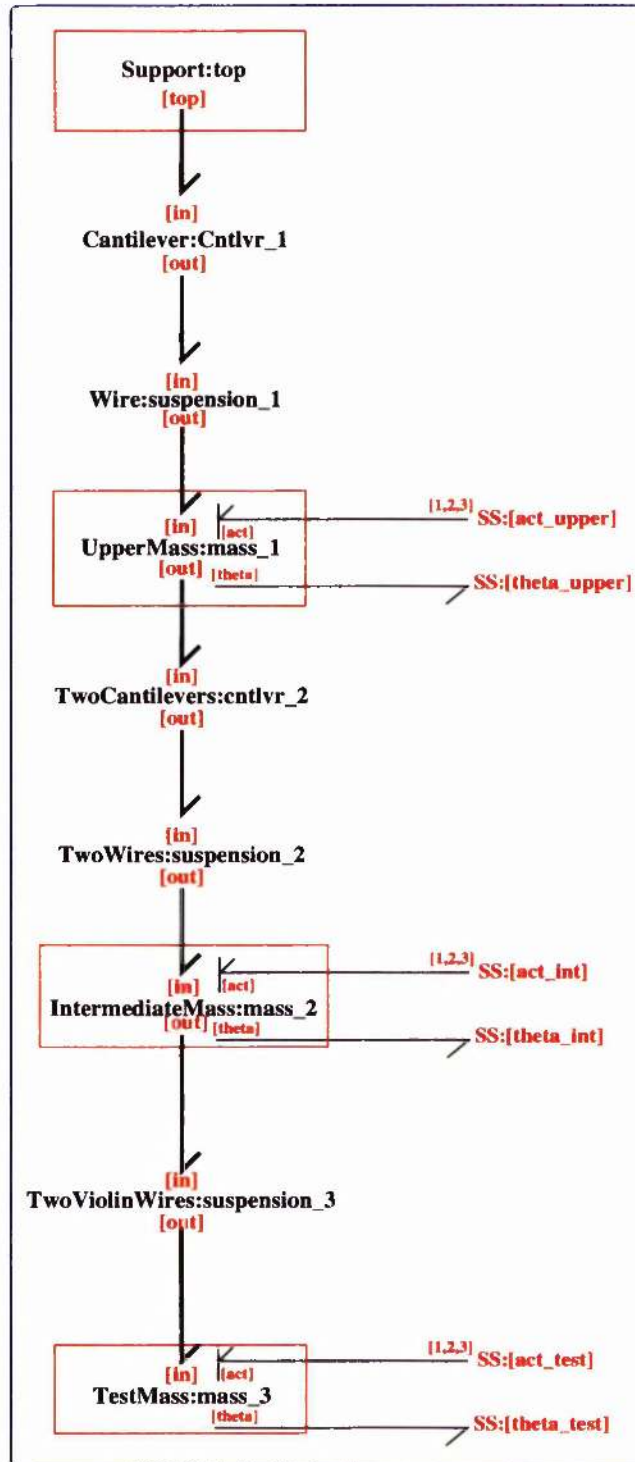


Figure 7.18: Bond graph of a GEO 600 gravitational wave detector suspension. Notice that this is a three stage pendulum with two stages of cantilever blades.

## Chapter 8

# Conclusions

In this thesis the requirement for multi-stage pendulum models was demonstrated, for which the bond graph domain was chosen. This is because the bond graph domain provides a unified approach to modelling hierarchical, unambiguous, core models, from which other representations can be generated. Furthermore, the construction of system models in a modular fashion facilitates the interchange of subcomponent models without the need to remodel the complete system. With this in mind a library of components was created. Validation of these components was achieved by creating the equivalent bond graph model of a physical system and comparing mode frequencies.

This thesis also showed that the bond graph domain offered a unified approach to system modelling and controller design. This was plainly borne out through the bond graph Model Based Observer (MBO). Here a bond graph model, of a real physical pendulum, was used in the synthesis of a bond graph MBO controller, designed to damp pendulum resonance. This MBO controller was implemented digitally and its performance compared with that of the original, classically designed, analogue controller to which it was shown that both controllers had a similar performance. However, since the “tuning” parameters of the MBO have physical meaning the design methodology is demonstrably more intuitive.

Having successfully created a MBO controller, for damping the pendulum modes, the potential of this design methodology was demonstrated with a MBO designed global controller.



Although this controller fell short of the very severe requirements of a GEO 600 global controller the proof of concept was shown. Here the split feedback topology, for a system with thirty-five states, was achieved with only three (effective) “tuning parameters”. Also, the (simulated) system response to step and impulse responses would, under less severe design goals, be deemed excellent.

## 8.1 Future work

Possible future work stemming from this thesis includes:

- The further development and implementation of the split feedback MBO.
- To prevent sensor noise from coupling into pendulum motion, sensors at the intermediate and upper masses, are actively filtered. Therefore, if a bond graph model of an operational amplifier were created (work in progress) then pendulum modelling, controller design and sensor filter design could all be undertaken in the bond graph domain. Moreover, previously designed analogue circuits could easily be created in the bond graph domain from which a suitable representation could be readily extracted for digital implementation.
- Finally, it is only natural that the 2-dimensional bond graph components, used in the modelling of pendula, within this thesis, should be extended to include the third spatial dimension.

## Appendix A

# Transforming the Bond Graph Representation of a Mass–Spring System into an Equivalent Transfer Function Representation

The annotated bond graph of this system can be seen in Figure A.1. Note, this is the causally complete bond graph of the acausal bond graph of Section 2.3 (Page 16). The causality has been completed assuming a velocity input and, following the rules outlined in Section 3.6 (Page 36). In this figure symbolic co-variables (efforts and flows) for each bond have been appended (see Section 3.7–Page 41 for an example of how to complete this stage). The set of ordinary differential equations which mathematically describe this mass spring system (with the causality shown) can easily be extracted from this bond graph, i.e.

$$\dot{x}_1(t) = u(t) - \frac{x_2(t)}{m} \quad (\text{A.1})$$

$$\dot{x}_2(t) = kx_1(t) + b \left( u(t) - \frac{x_2(t)}{m} \right) - 0 \quad (\text{A.2})$$

$$y(t) = \frac{x_2(t)}{m} \quad (\text{A.3})$$

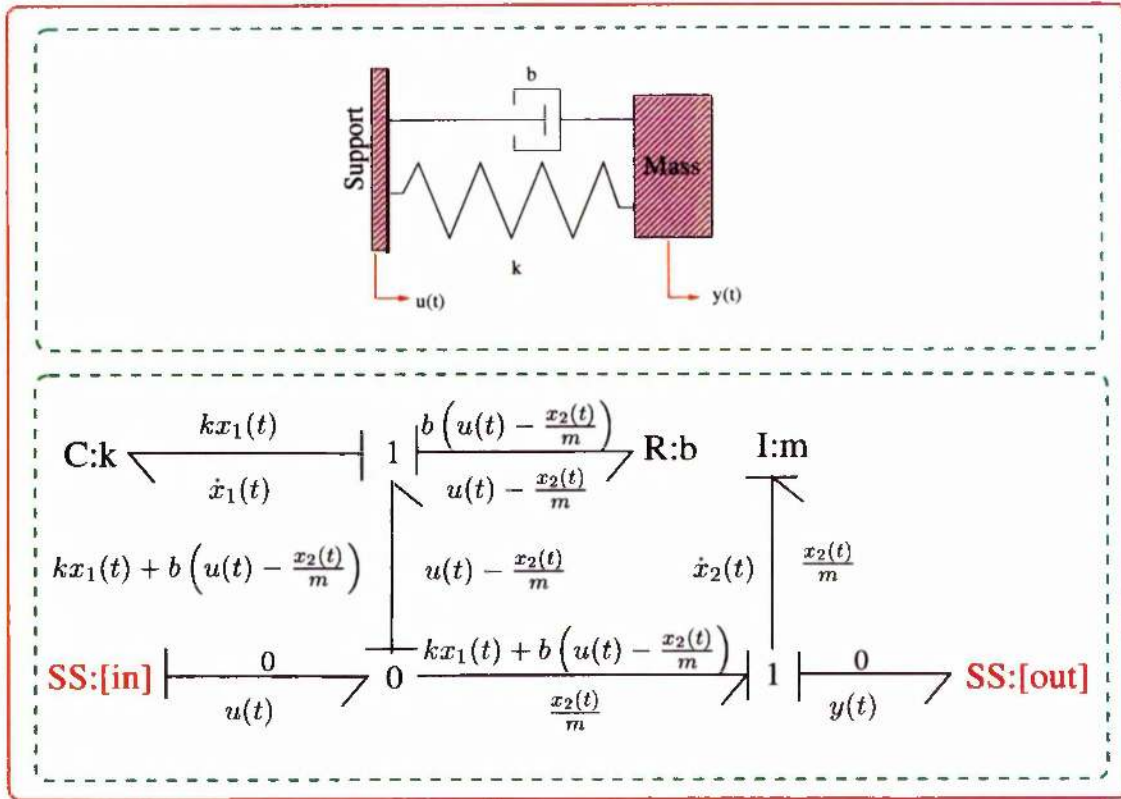


Figure A.1: The schematic and causal bond graph of a Mass-Spring System

The state matrices (A,B,C,D) of the state space representation

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (\text{A.4})$$

$$y(t) = Cx(t) + Du(t) \quad (\text{A.5})$$

$$(\text{A.6})$$

are then simply:

$$A = \begin{bmatrix} 0 & -\frac{1}{m} \\ k & -\frac{b}{m} \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ b \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & \frac{1}{m} \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \end{bmatrix}$$

where  $x(t)$  is a column vector of the states and  $u(t)$  is a vector of inputs i.e.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and

$$u = [u_1]$$

Finally the transfer function  $G(s)$  is generated from  $C[sI - A]^{-1}B + D$ , ( $I$  is the identity matrix of the same dimensions as  $A$ ), giving:

$$G(s) = \frac{bs + k}{ms^2 + bs + k} \quad (\text{A.7})$$

Now it should be appreciated that the causal bond graph representation is an unambiguous representation of a system, and that each stage of the transformation from bond graph to transfer function is a simple “turning of the handle” process. Hence these stages, including the possible completion of causality (for given input causality) can be undertaken by a suitable computer program.

## Appendix B

# Introduction to Bicausal Bond Graphs

In Chapter 6 it was seen that *bicausal* bond graphs were utilised in modelling the inverse dynamics of a sensor. That is, the bond graph sensor models an idealised instrument whose input measures and integrates velocity to give displacement (sensor output). The task then, is to model the inverse of this process such that velocity is derived from the differentiation of the sensor output. The bicausal bond graph model of this sensor inversion can be seen in Figure B.1. Firstly, note that this bond graph component is hierarchical, containing an instance of the **INTF** component which was used to model the sensor (Section 6.5.1). Secondly, two bicausal **SS** components can be seen, identified by the fact that half-stroke, rather than full stroke, causality has been applied. The **SS** component **SS:z** provides zero effort and flow sources and **SS:v** provides effort and flow sensors (remember that **SS** components with full causal stroke assignment, provide either an effort source and flow sensor or a flow source and effort sensor).

Prior to presenting an annotated bond graph describing the inverse dynamics of a sensor the basic theory of bicausal bond graphs will be presented.



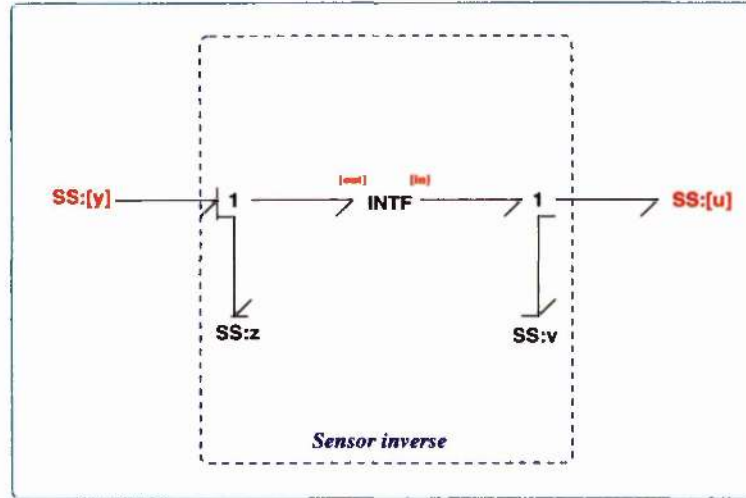


Figure B.1: *Hierarchical bond graph component utilising bicausal bonds to model the inverse dynamics of inverse sensor dynamics*

## B.1 Bicausal Bond Graphs

Bicausal bond graphs were introduced by Gawthrop [61] to give a foundation for deriving system properties relating to system inversion, state estimation and parameter estimation directly from the system bond graph [62]. In this appendix only the system inversion aspects of bicausal bond graphs is presented. For further reading see, [62] [61].

### B.1.1 Bicausal Bonds

The two bicausal bonds can be seen, along with their associated assignment equations, in Figure B.2. The notation is such that:

- a *causal half-stroke* on the flow side of the bond (the common convention is that half-arrow head is on the flow side of a bond) implies that *flow* is imposed on the variable at the *far* of the bond (the variable associated with the far end of the bond is on the left-hand side of the assignment statement) and
- a *causal half-stroke* on the *effort* side of the bond implies that *effort* is imposed on

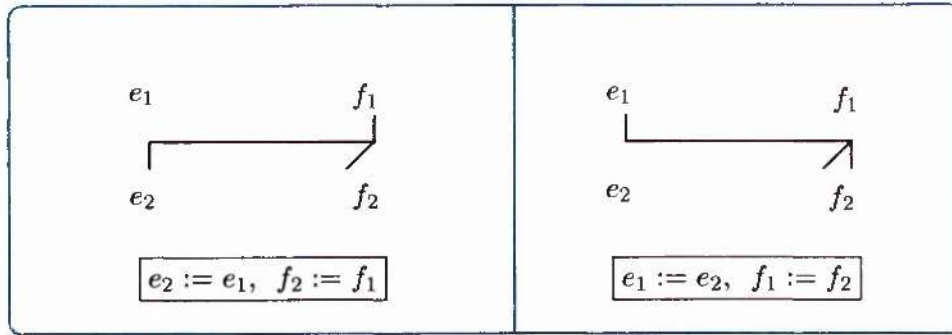


Figure B.2: *Bicausal bonds*

the variable associated with the *near* end of the bond (the variable associated with the near end of the bond is on the left-hand side of the assignment statement).

This notation has the result that when causal half-strokes are applied at the same end of a bond they have the same meaning as the corresponding (full) causal stroke (Section 3.6).

### B.1.2 Bicausal Junctions

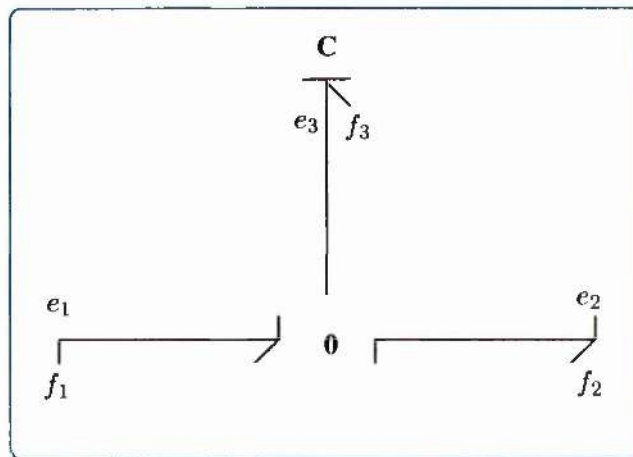


Figure B.3: *The propagation of bicausality at a junction*

Bicausality can be propagated through junctions, an example is shown in Figure B.3. Here the **C** component is assumed known and is thus unicausal. Now, given the bicausality of

bond 1 ( $e_1$  and  $f_1$ ) the impinging effort ( $e_1$ ), at the junction, forces the causality on the **C** component<sup>1</sup> (i.e it is forced to have derivative causality). Thus the causality of flows  $f_1$  and  $f_3$ , in conjunction with the constitutive relationship for the **0** junction, implies the causality of  $f_2$ , as shown.

### B.1.3 Source Sensor SS Components

The **SS** component was introduced in Section 3.4.1. The four bicausal forms of this component can be viewed in Figure B.4. Notice the first two forms have their half-strokes at the same end of their respective bonds and hence have the same properties as the full causal **SS** components of Section 3.4.1, namely:

- (1) effort source and flow sensor
- (2) flow source and effort sensor

whereas the two bicausal **SS** components with half strokes at opposite ends of their respective bond are

- (3) flow and effort source
- (4) flow and effort sensor

respectively.

### B.1.4 Flow and Effort Amplifiers

The *flow amplifier* (**AF**) and *effort amplifier* (**AE**) are ideal two port flow and effort amplifiers respectively. With the standard causal pattern, as shown in Figure B.5, the following set of equations,

$$e_1 := 0, \text{ and } f_2 := f_1 \quad (\text{B.1})$$

---

<sup>1</sup>Remember at a **0** junction there is common effort, and hence only one bond can impose effort and also that  $\sum \text{flows} = 0$

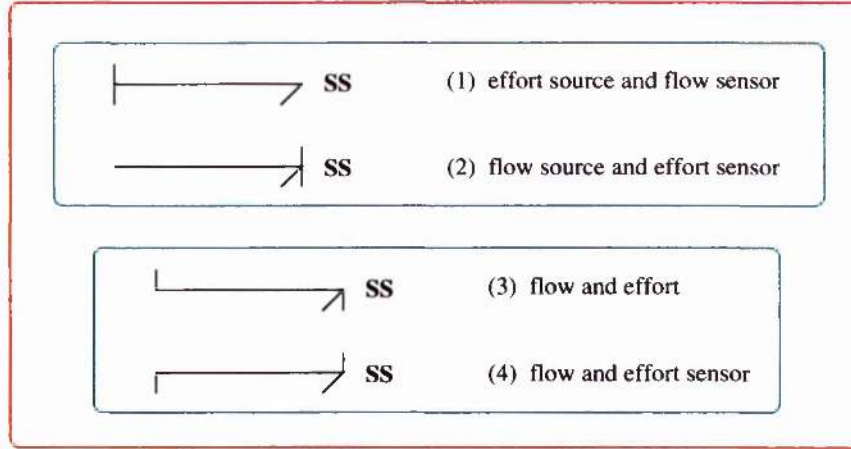


Figure B.4: *Bicausality and source sensor components SS*

for the **AF** component and

$$f_1 := 0, \text{ and } e_2 := e_1 \quad (\text{B.2})$$

for the **AE** component, apply. However, if the output flow ( $f_2$ ) of the **AF** component is known, then the bicausal half-strokes as shown in Figure B.5 apply, resulting in the following assignment statements:

$$e_1 := 0, \text{ and } f_1 := f_2 \quad (\text{B.3})$$

Similarly if the output effort ( $e_2$ ) of the **AE** component is known then the bicausal half-strokes as shown in Figure B.5 apply, resulting in the following assignment statements:

$$f_1 := 0, \text{ and } e_1 := e_2 \quad (\text{B.4})$$

The bicausal bond graph has been extended to incorporate other components [62]. These are not covered here.

## B.2 Using Bicausal bond graphs to model the inverse dynamics of a sensor

In Figure B.6 the annotated bond graph component that models the inversion of the sensor dynamics can be seen. The determination of each bond's co-variables was simply deter-

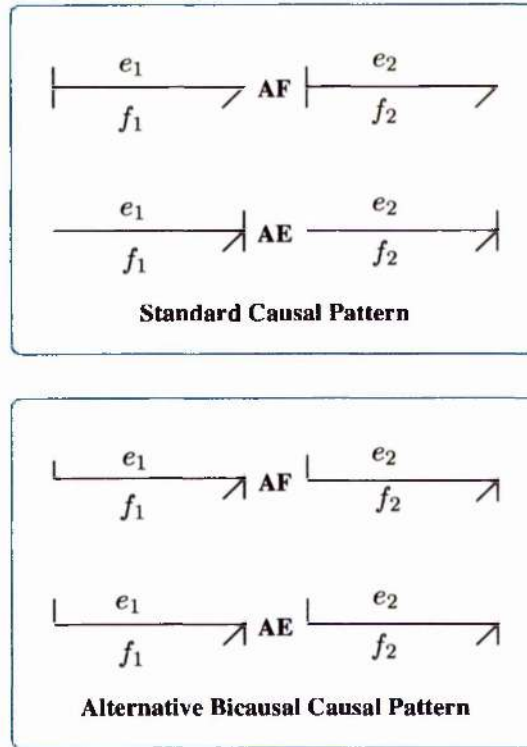


Figure B.5: Possible bicausal assignments as applied to flow and effort amplifiers

mined from the definitions presented above and in Chapter 3. Here, for clarity, this bond graph model has been presented as a non hierarchical system (c.f. Figure B.1). Points to note are;

- The component **SS:z** provides zero effort and flow sources. The zero effort sources, through the propagation of causality, imposes derivative causality on the **C** component. The zero flow source in conjunction with the **AE** component ensures that there is no “back effect” on the system whose effort variable is to be differentiated.
- The **AF** is simply included so that only the derivative  $\dot{q}$  is propagated to any external components.
- The **1** junction with the attached bond graph component **SS:v** is simply included to provide effort and flow sensors. The effort sensor carries the effort input from the **SS:[u]** (i.e the effort variable associated with a conjoined component). The flow



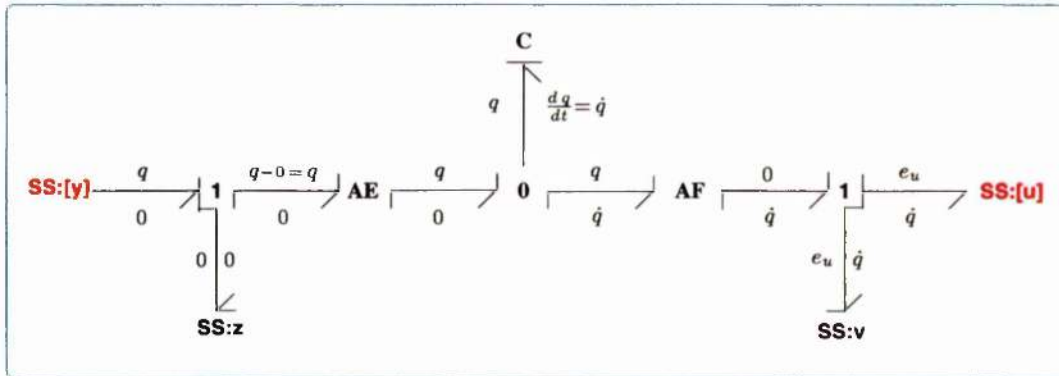


Figure B.6: *The annotated bond graph of the bond graph component modelling inverse sensor dynamics*

sensor outputs the derived velocity. Furthermore, the use of this bicausal **SS** component ensures that the output port (**SS:[u]**) has a full causal stroke.

Notice that, as desired, this bond graph model takes an effort variable as its input and provides an output (flow) which is the derivative of the input.

## Appendix C

# Split feedback Numerical Parameters

Here are the numerical parameters as used in the design of the global controller

```
% *****
g      = 9.81;
% *****
ux      = 0.1;          % dimensions of upper mass (square)
uy      = 0.3;
uz      = 0.07;
den1    = 2700;          % density (aluminium)
m_1     = den1*uy*uz*ux; % mass
%ix     = m_1*(uy^2+uz^2)/12; % moment of inertia (sideways roll)
j_1     = m_1*(uz^2+ux^2)/12; % moment of inertia (longitud tilt)
%iz     = m_1*(uy^2+ux^2)/12; % moment of inertia (rotation)
% *****
ix      = 0.1;          % dimension of int mass (cylinder)
ir      = 0.09;
den2    = 2202;          % density (fused silica)
m_2     = den2*pi*ix^2*ix; % intermediate mass
%ix     = m_2*(ir^2/2); % moment of inertia (sideways roll)
j_2     = m_2*(ir^2/4+ix^2/12); % moment of inertia (longitud tilt)
%iz     = m_2*(ir^2/4+ix^2/12); % moment of inertia (rotation)
% *****
```

Table C.1: Numerical parameters used in global control design

Continued on subsequent pages...

```

%*****
tx      = 0.1;                % dimensions of test mass (cylinder)
tr      = 0.09;
den3    = 2202;                % density (fused silica)
m_3     = den3*pi*tr^2*tx;     % test mass
%I3x    = m_3*(tr^2/2);        % moment of inertia (sideways roll)
j_3     = m_3*(tr^2/4+tx^2/12); % moment of inertia (longitud tilt)
%I3z    = m_3*(tr^2/4+tx^2/12); % moment of inertia (rotation)
%*****
l_1     = 0.42;                % upper wire length
l_2     = 0.187;               % intermediate wire length
l_3     = 0.280;               % lower wire length
%*****
r_1     = 350e-6;               % radius of upper wire
r_2     = 175e-6;               % radius of intermediate wire
r_3     = 154e-6;               % radius of lower wire
%*****
y_1     = 1.65e11;              % youngs mod of wire 1 (s/steel 302)
y_2     = 1.65e11;              % youngs mod of wire 2 (s/steel 302)
y_3     = 7e10;                 % youngs mod of wire 3 (f silica)
%*****
% spring constants

k_1     = 2*y_1*pi*r_1^2/l_1    % spring constant for wires (susp 1)
k_2     = 2*y_2*pi*r_2^2/l_2    % spring constant for wires (susp 2)
k_3     = 2*y_3*pi*r_3^2/l_3;   % spring constant for wires (susp 3)

k_c1    = 2*739;                % effective spring const cantil 1
k_c2    = 2*498;                % effective spring const cantil 2
% *****
% parameters for violin modes - suspension 3 only *****
rho_w    = 2202;                % density of steel
a_w      = pi*r_3^2;            % x-sectional area of wire
n_wires  = 4;                   % number of wires
n_lumps  = 4;                   % number of lumps
dx_wires = l_3/n_lumps;
mass     = m_3;
wire_tension = (mass*g)/n_wires;
lin_rho  = rho_w*a_w;
dk_3     = (wire_tension/dx_wires)
dm_3     = dx_wires*lin_rho
dr_3     = 0.0;
%*****

```

Table C.2: Numerical parameters used in global control design continued

```

% suspension locations
vu_1 = 0.001; % Upper Mass Susp
vdl_1 = -0.001; % Upper Mass Susp
vdr_1 = -0.001; % Upper Mass Susp
vul_2 = 0.001; % Intermediate Mass Susp
vur_2 = 0.001; % Intermediate Mass Susp
vdl_2 = -0.001; % Intermediate Mass Susp
vdr_2 = -0.001; % Intermediate Mass Susp
vul_3 = 0.001; % Test Mass Susp
vur_3 = 0.001; % Test Mass Susp
hu_1 = 0.0; % Upper Mass Susp
hdl_1 = -0.03; % Upper Mass Susp
hdr_1 = -0.03; % Upper Mass Susp
hul_2 = 0.03; % Intermediate Mass Susp
hur_2 = 0.03; % Intermediate Mass Susp
hdl_2 = -0.005; % TestMassSusp,Triple
hdr_2 = -0.005; % TestMassSusp,Triple
hul_3 = 0.005; % Test Mass Susp
hur_3 = 0.005; % Test Mass Susp
%*****
% actuator locations
h_ai_1 = -ix/2; % Actuator 1 Intermediate Mass
h_ai_2 = -ix/2; % Actuator 2 Intermediate Mass
h_at_1 = -tx/2; % Actuator 1 Test Mass
h_au_1 = -ux/2; % Actuator 1 Upper Mass
h_au_2 = -ux/2; % Actuator 2 Upper Mass
v_ai_1 = ir/2; % Actuator 1 Intermediate Mass
v_ai_2 = -ir/2; % Actuator 1 Intermediate Mass
v_at_1 = 0.000; % Actuator 1 Test Mass
v_au_1 = uz/2.0; % Actuator 1 Upper Mass
v_au_2 = -uz/2.0; % Actuator 2 Upper Mass
%*****
% **** These parameters can be used to normalise inputs/outputs
norm_i_1 = 1.0; % DcContObs4 Actuator 1 Intermediate Mass
norm_i_2 = 1.0; % DcContObs4 Actuator 2 Intermediate Mass
norm_t_1 = 1.0; % VelContObs Actuator 1 Test Mass
norm_u_1 = 1.0; % DcContObs4 Actuator 1 Upper Mass
norm_u_2 = 1.0; % DcContObs4 Actuator 2 Upper Mass
wnorm = 1.0;
%*****
r_ou_1 = 1e5; % compensator stuff
r_ou_2 = 1e5; %
r_oi_1 = 1e5; % Observer Gains
r_oi_2 = 1e5; %
r_ot_1 = 1e5; %
r_cu_1 = 60.0; % Upper Mass Damping Coefficient (Act 1)
r_cu_2 = 60.0; % Upper Mass Damping Coefficient (Act 2)
r_ci_1 = 35.0; % Int Mass Damping Coefficient (Act 3)
r_ci_2 = 35.0; % Int Mass Damping Coefficient (Act 4)
r_ct_1 = 55.0; % Test Mass Damping Coefficient (Act 5)
r_p = 40; % Proportional Action
c_i = 100; % Integral Action (Intermediate Mass)
c_u = 100; % Integral Action (Upper Mass)

```

Table C.3: Numerical parameters as used in global control designed continued

# Bibliography

- [1] Courtesy of Dr. G. Cagnoli, Glasgow University.
- [2] Courtesy of Dr. M. Plissi, Glasgow University.
- [3] D. C. Karnop D. L. Margolis and R. C. Rosenberg. *System Dynamics. Modelling and Simulation of Mechatronic Systems*. Wiley-Interscience Publications, 2000.
- [4] W. Forsythe and R. M. Goodall. *Digital Control*. Macmillan New Electronics Series, 1991.
- [5] A. Einstein. Die grundlage der allgemeinen relativitatstheorie. *Annalen der Physik*, 49:769, 1916.
- [6] J. A. Lobo. Sources of gravitational waves in general relativity. In *Proceedings of the 46<sup>th</sup> Scottish Universities Summer School in Physics*, 1995.
- [7] J. Hough et al. Design study report, a British long baseline gravitational wave observatory. *GWD/RAL/86-001*, 1986.
- [8] B. F. Schutz. The detection of gravitational waves. In *1995 Houches School on Astrophysical Sources of Gravitational Radiation*, 1996.
- [9] K. Thorne. Gravitational waves. In *Snowmass 95 Summer Study on Partical and Nuclear Astrophysics and Cosmology*, 1996.
- [10] J. H. Taylor and J. M. Weisberg. A new test of general relativity: Gravitational radiation and the binary pulsar psr1913+16. *The Astrophysical Journal*, 253, 1982.



- [11] J. Weber. Detection and generation of gravitational waves. *Physics Review*, 117, 1960.
- [12] Astone P. Long term operation of the Rome Explorer cryogenic gravitational wave detector. *Physics Review*, 47(2):362–375, 1993.
- [13] Z. K. Geng, W. O. Hamilton, W. W. Johnson, E. Mauceli, S Merkowitz, A. Morse and N. Solomonson. Operation of the ALLEGRO Detector at L.S.U. In *First Eduardo Amaldi Conference on Gravitational Wave Experiments*. World Scientific Singapore, 1999.
- [14] G. V. Pallottino. The cryogenic gravitational wave antennas Explorer and Nautilus. In *Proceedings of VIRGO 96 (Cassina)*. World Scientific, Singapore, 1996.
- [15] B. Price, B Xu N. Soloman, O. Aguiar, Z. Geng, W. O. Hamilton, W. W. Johnson, S. Merkowitz and N. Zhu. The 1991 LSU Gravitational radiation detector: Modifications, performance and prospects. In *Proceedings of the Sixth Marcel Grossman Meeting on General Relativity*. World Scientific, Singapore, 1992.
- [16] W W Johnson and S M Merkowitz. Truncated icosahedral gravitational wave antenna. *Physics Review Letter*, 70, 1993.
- [17] <http://sam.phys.lsu.edu/tiga>.
- [18] A. D. Ward. MiniGRAIL a Small (60cm) Spherical Antenna. In *Third Eduardo Amaldi Conference on Gravitational Wave Experiments*. World Scientific Singapore, 1995.
- [19] B. J. Meers. Recycling a laser–interferometric gravitational– wave detectors. *Physics Review D*, 38, 1988.
- [20] R. W. P. Drever. Gravitational radiation. In N Derouille and T Piran, editors, *Les Hucchees Summer Institute*, 1988.
- [21] H. B. Callen and T. A. Welton. Irreversible and generalised noise. *Physics Review*, 83(1), 1951.
- [22] H. B. Callen and R. F. Greene. On a theorem of irreversible thermodynamics. *Physics Review*, 86(5), 1952.

- [23] S. M. Twyford. *Developments Towards Low Loss Suspensions for Laser Interferometric Gravitational Wave Detectors*. PhD thesis, University of Glasgow, 1998.
- [24] J Hough. *The Detection of Gravitational Waves*. Cambridge University Press, 1991.
- [25] A Rüdiger, R Schilling, W. Winkler, H. Biling, and K. Maischberger. A mode detector to suppress fluctuations in laser beam geometry. *Optica Acta*, 28:641, 1981.
- [26] K. Skeldon, K Strain, A Grant, J Hough. Test of an 18-m-long suspended mode cleaner cavity. *Review of Scientific Instruments*, 67(7), 1996.
- [27] M. V. Plissi, C. I. Torrie, M. E. Husman, N. A. Robertson, K.A. Strain, and H. Ward. GEO 600 triple pendulum suspension system: Seismic isolation and control. *Review of Scientific Instruments*, 71(6):2539-2545, June 2000.
- [28] A. L. Kimball and D. E. Lovell. Internal friction in solids. *Physics Review*, 30, 1927.
- [29] S. M. Twyford. Developments towards low loss suspensions for laser interferometric gravitational wave detectors. PhD Thesis, 1998.
- [30] C. T. E. Torrie. *Development of Suspensions for the GEO 600 Gravitational Wave Detectors*. PhD thesis, University of Glasgow, 1999.
- [31] M. V. Plissi C, Torrie, N. A. Robertson, S. Killbourne, S. Rowan, S. M. Twyford, H. Ward, K. Skeldon, and J. Hough. Aspects of the suspension system for GEO 600. *Review of Scientific Instruments*, 69(8), 1998.
- [32] J. Hough, H Walter B.F Shultz, J. Eulers, H. Welling, I Corbett, V. Kose, et. al. *Proposal for a joint German-British Interferometric Gravitational wave detector*. Max-Planck-institute for Quantenoptik Report 147 and GWD/137/JH(89), 1989.
- [33] M. E. Husman. *Suspension and Control For Gravitational Wave Detectors*. PhD thesis, University of Glasgow, 1999.
- [34] M. Bernardini et. al. M. Beccaria. The creep problem in the virgo suspensions: a possible solution using maraging steel. *Nuclear Instruments & Methods in Physics A*, 404, 1998.

- [35] M. V. Plissi, C. Torrie, M. E. Husman, N. A. Robertson, K. A. Strain, H. Ward, H. Luck, and J Hough. GEO 600 triple pendulum system: Seismic isolation and control. *Review of Scientific Instruments*, 71(6), 2000.
- [36] G I Gonzalez P R Saulson. Brownian motion of a mass suspended by an anelastic wire. *Journal of the acoustical society of America*, 96:207–212, April 1994.
- [37] P. J. Gawthrop and L. P. S. Smith. *Metamodelling: Bond Graphs and Dynamic Systems*. Prentice Hall, Hemel Hempstead, Herts, England., 1996.
- [38] P Wellstead. *Introduction to Physical System Modelling*. Academic Press, London, 1979.
- [39] [www.eng.gla.ac.uk/bg.html](http://www.eng.gla.ac.uk/bg.html).
- [40] [www.sourceforge.net/projects/mtt](http://www.sourceforge.net/projects/mtt).
- [41] [www.eng.gla.ac.uk/bg/bg\\_soft.html](http://www.eng.gla.ac.uk/bg/bg_soft.html).
- [42] [www.bondgraphs.com/software](http://www.bondgraphs.com/software).
- [43] D. Karnopp. Power conserving transformers: Physical interpretations and applications using bond graphs. *Journal of the Franklin Institute*, 288(3):175–201, September 1969.
- [44] Dean Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. *System Dynamics : Modeling and Simulation of Mechatronic Systems*. Horizon Publishers and Distributors Inc, 3rd edition, January 2000.
- [45] T. Kailath. *Linear Systems*. Prentice–Hall, New York, 1980.
- [46] [www.mech.gla.ac.uk/~dpalmer/pendulum\\_components](http://www.mech.gla.ac.uk/~dpalmer/pendulum_components).
- [47] [www.che.wisc.edu/octave](http://www.che.wisc.edu/octave).
- [48] S. Kilbourn. *Double Pendulums for Terrestrial Interferometric Gravitational Wave Detectors*. PhD thesis, University of Glasgow, 1997.
- [49] P M Morse. *Vibrations and Sound*. McGraw-Hill, 1948.

- [50] D. L. Margolis. *Bond Graphs for Engineers*, chapter Application of Bond Graphs to Distributed Parameter Systems. Elsevier Science Publishers, Amsterdam, 1992.
- [51] P. J. Gawthrop and L. Smith. Causal augmentation of bond graphs. *Journal of the Franklin Institute*, 329(2):291–303, 1992.
- [52] H. J. Pain. *The Physics of Vibrations and Waves*, page 103ff. Wiley, 93.
- [53] G. S. Vink. *Digital Computer Control Systems*. Prentice–Hall International Editions, 1991.
- [54] G. F. Franklin and J. Powell. *Digital Control of Dynamic Systems*. Addison–Wesley Publishing Company, 1980.
- [55] Tustin. A method of analysing the behaviour of linear systems in terms of time series. *Joint Institute for Energy and Environment*, 94:130–142, 1947.
- [56] C. L. Phillips and H. T. Nagle. *Digital Control System Analysis and Design*. Prentice–Hall, New York, 1990.
- [57] B. Lin. The effect of finite word length on the accuracy of digital filters - a review. *IEEE Trans. on Circuit Theory*, CT18(6):670 – 677, 1971.
- [58] P. J. Gawthrop. Physical model based control: A bond graph approach. *Journal of the Franklin Institute*, 332B:285–305, September 1995.
- [59] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley, 1972.
- [60] S. Rowan. *Aspects Of lasers for the Illumination Of Interferometric Gravitational Wave Detectors*. PhD thesis, University of Glasgow, 1995.
- [61] P. J. Gawthrop. Bicausal bond graphs. In F. E. Cellier and J. J. Granda, editors, *Proceedings of the International Conference On Bond Graph Modeling and Simulation (ICBGM'95)*, volume 27 of *Simulation Series*, pages 83–88, Las Vegas, U.S.A., January 1995. Society for Computer Simulation.
- [62] P. J. Gawthrop. Physical interpretation of inverse dynamics using bicausal bond graphs. *Journal of the Franklin Institute*, 337, 2000.

